SECTION: INFORMATION TECHNOLOGY AND CYBERSECURITY

AN APPROACH TO USING MOBILENET CNN-MODEL FOR GESTURE RECOGNITION

Mazurets Oleksandr
Ph.D (Engineering Science), Associate Professor
exe.chong@gmail.com
Zalutska Olha
Postgraduate student
zalutskolha@gmail.com
Tyschenko Olena
Teacher
tyschenko.helen@gmail.com
Bohdanova Anhelina
Bachelor student
gelya.bogdanova.88@gmail.com
Computer Science Department
Khmelnytskyi National University, Ukraine

Statistics show that the risk of hearing or speech problems is only increasing, not to mention genetic rather than acquired diseases. According to WHO experts, by 2050, more than 900 million people will suffer from hearing loss in one way or another, taking into account various factors, both genetic and environmental, and the number of people with visual impairments is growing by 1 million people every year. More than 2 million people live with hearing impairments in Ukraine, and almost 12,000 people are recognized as disabled due to visual impairments in Ukraine.

Society often does not take into account their everyday needs, and almost always – the right to a multicultural life. They do not have the opportunity to visit theaters, philharmonics, museums and live an ordinary life like other people [1].

In order for people with certain impairments to be able to better express their thoughts with the help of their limited resources, the goal of this course project was to create a neural network that could learn to recognize the gestures of a specific person who is currently using the created software product, in order to better recognize certain signs shown by the user [2]. Gesture recognition will be useful not only for people with visual and hearing impairments, but also for people with other disabilities, such as muscle or joint problems.

To solve the task – recognition of images (gestures), such architectures of neural networks as multilayer perceptron, recursive neural networks, networks of long short-term memory, as well as convolutional neural networks – Convolutional neural nets (CNN) have proven themselves well. It is this architecture that will be used in this

course project to solve the problem of real-time gesture recognition, because it allows optimal use of memory for memorizing information and therefore does a better job of recognizing high-quality pictures.

In the future, this approach can be improved and refined to recognize gestures and translate them into text format. In this case, the created neural network will also be useful for those people who suffer from carpal tunnel syndrome, Parkinson's disease or various manifestations of arthritis or arthrosis, which is increasingly common in young people, because then the interaction with the keyboard and mouse will be minimized. that can facilitate the daily life of people with such diagnoses.

The aim of the work is to develop an approach for gesture recognition using the CNN neural network of the MobileNet model in real time.

For gesture recognition using the real-time neural network gesture recognition method, it was decided to use such a neural network architecture as CNN – a convolutional neural network, namely the MobileNet model presented by the TensorFlow.js library.

The MobileNet model is designed for use in mobile applications and is TensorFlow's first mobile computer vision model.

MobileNet uses depth-separated convolutions. This significantly reduces the number of parameters compared to a network with regular convolutions of the same depth. This leads to the use of a simplified version of deep neural networks, but no less effective, in particular, in the recognition of visual images.

MobileNet is Google's open-source class of ANNs, and it provides an excellent starting point for training classifiers and solving classification problems [3].

Split-depth convolution consists of two operations (Figure 1):

– Deep convolution.
– Point convolution.

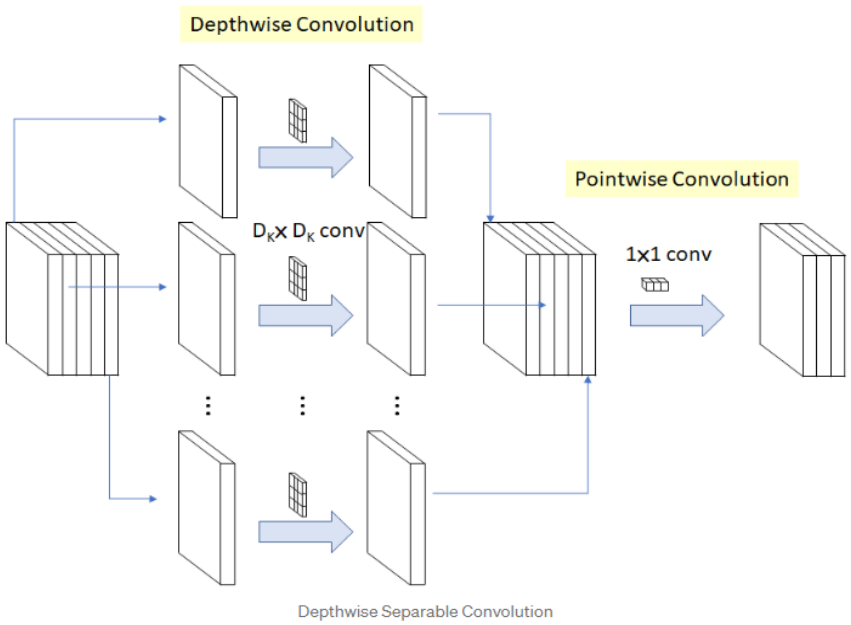

Figure 1. General diagram of the MobileNet model [3].

Depth convolution arose from the idea that the depth and spatial dimensions of a filter can be separated – hence the name separated (Figure 2).

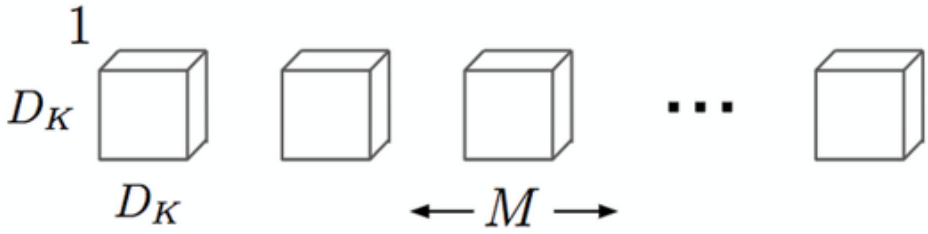

Figure 2. Depth convolution for one input channel [3].

In this project, a 244x244 real-time webcam captured image is transferred into the model, so there are 244x244 input channels. Accordingly, the number of output channels is the same as the number of input channels, i.e., in this case – 244 channels.

A point convolution is a convolution with a channel size of 1x1, that is, it combines the layers created using a depth convolution (Figure 3).

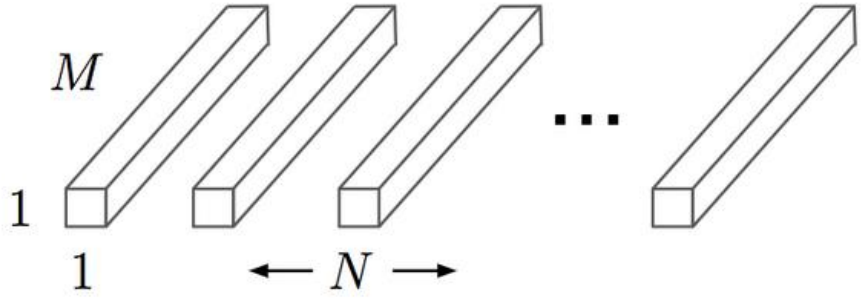

Figure 3. Point convolution [3].

Let's consider the model in more detail and the necessary steps for obtaining data, training the network and obtaining output data – the result of recognizing the gestures of the current user. After receiving real-time data – images, it is necessary to normalize the received data so that the model can work with them in the future. For this, the image normalization function provided by the TensorFlow.js library is used. Next, the normalized data is transferred to and stored in the MobileNet model.

After the model has received all the input data, it is necessary to conduct a training phase of the model so that it can perform the function of gesture recognition. This requires first encoding the normalized input data as an OH-vector. An OH-vector is a unitary code, a group of bits, among which the only allowed combinations of values are those in which only one bit is set (1) and all others are off (0). These vectors are used as target labels during model training.

Next, you need to sequentially pass the input data through several inner layers, such as the Flatten and Dense layers.

The Flatten layer is used to convert input data into a smaller dimension, that is, into a one-dimensional vector [4].

The next layer is the Dense layer, the activation function of which is the RELU function. This is one of the most popular features for deep learning. The logic of the

function is as follows: the RELU function returns 0 if it receives a negative argument and the number itself if it receives a positive argument.

The RELU function can be represented by the following formula:

$$f(z) = \max(0, z), \quad (1)$$

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

The next step is to transfer the data back to the Dense layer, but with the activation function SOFTMAX, which transforms the vector *z* of dimension *K* into the vector *σ* of the same dimension, where each coordinate σ_i of the resulting vector is represented by a real number in the interval [0, 1] and the sum of the coordinates is 1. Coordinates σ_i are calculated as follows [5]:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}, \quad (2)$$

where *z* is the input vector, *K* is the dimension of the vector, *σ* is another vector of the same dimension.

Coordinates σ_i of the received vector are interpreted as the probability that the object belongs to class *i*.

In this case, the vector-column *z* is calculated as follows [5]:

$$z = w^T x - \theta, \quad (3)$$

where *x* is a column vector of features of an object of dimension $M \times 1$, w^T – *is* a transposed matrix of weight coefficients of features of an object of dimension $K \times M$, θ is a column vector with a threshold value of dimension $K \times 1$, *K* – the number of classes of the object, *M* – the number of features of the object.

The Adam Optimizer [6] function is used as an optimizer:

$$w_{t+1} = w_t - \alpha w_t, \quad (4)$$

$$m_t = \beta w_{t-1} + (1 - \beta)\left[\frac{\delta L}{\delta w_t}\right], \quad (5)$$

where m_t – is the set of gradients at time *t* [current time] (initially $m\_t = 0$), *m(t-1)* set of gradients at time t-1 [previous time], w_t – weights at time t, $w_{(t+1)}$ –weights at time *t+1*, α_t – learning rate at time *t*, δL – derivative of the error function, δw_t – derivative of weights at time *t*, β – parameter of the moving average (const, 0,9).

Also, the categoricalCrossentropy function is used to determine the error between the expected output data and the real one, the function returns the value in the form of an OH-vector.

To check the correctness of the proposed approach, an information system was created [7]. To start using the application, you need to launch the application, then the user will see the program interface in front of him. When the video is synchronized, you can start showing the gestures. To do this, you need to show a certain gesture, in this case it is a number from 0 to 5 and press the corresponding button. This step can be performed 1 time for the model to learn to recognize 1 digit, or for each digit. The number of images created will be displayed to the right of the button with the corresponding number (Figure 4).

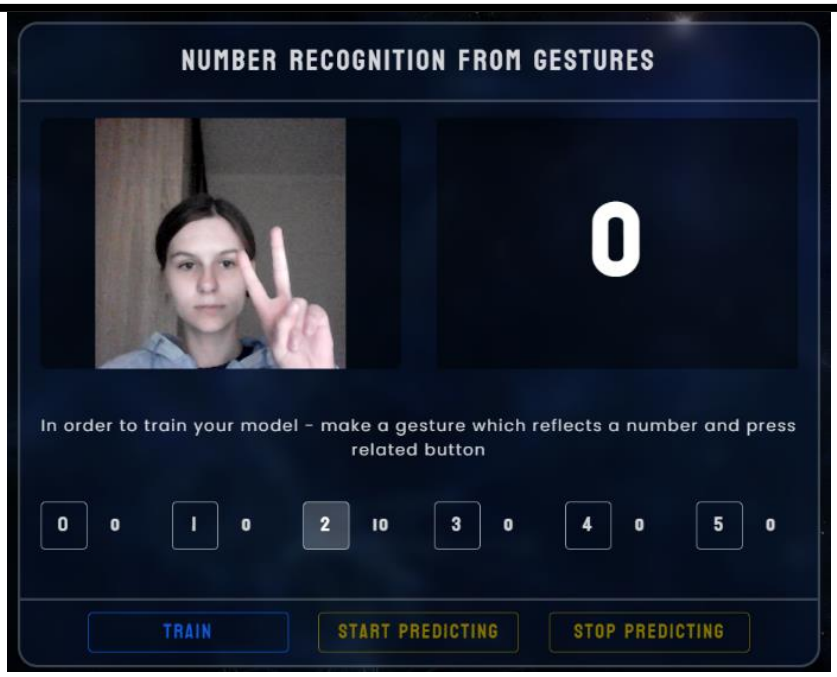

Figure 4. Showing the gesture for further training of the model

When the user has finished the demonstration of gestures, it is necessary to click on the “Train” button so that the model starts learning to recognize the user's gestures during subsequent demonstrations and wait for the message that the model is ready for use (Figure 5).

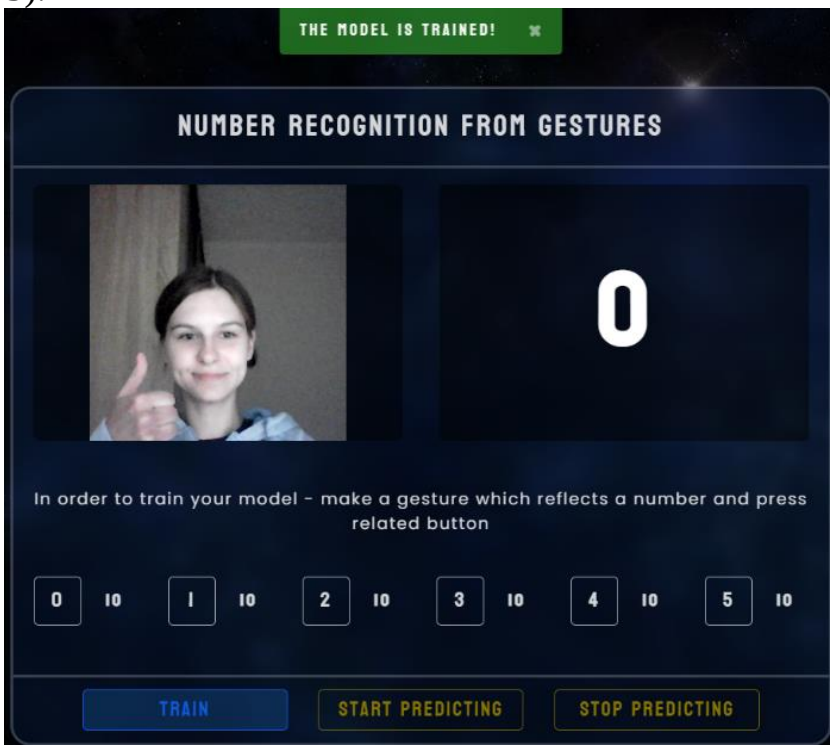

Figure 5. The result of model training

To recognize gestures, you need to click on the “Start predicting” button, show the gesture, and the neural network will recognize the user's gesture, which will be displayed to the right of the user's camera (Figure 6).

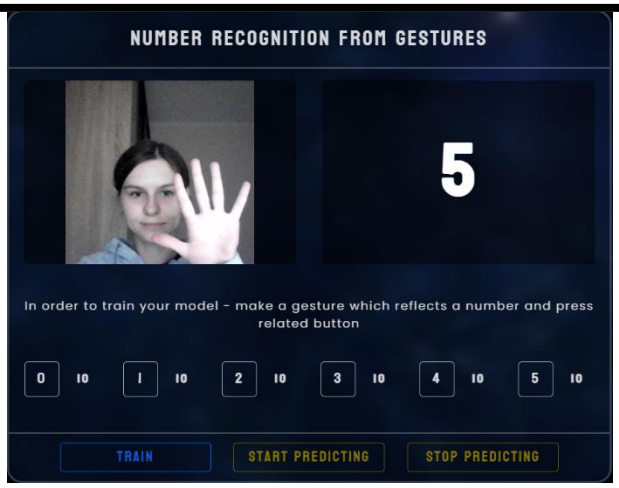

Figure 6. The result of gesture recognition by a neural network

As a result of the work performed, one of the types of MobileNet architecture of CNN artificial neural networks was applied in practice to solve the problem of gesture recognition in real time. The obtained results are important not only in the scientific sense, but also for practical application, because neural networks help people with various diseases, such as Parkinson's disease, speech or hearing impairment, tunnel syndrome, to facilitate their interaction with the computer, to carry out effective studying in schools and universities and, of course, socializing. So, these are just the first steps to simplify people's lives as much as possible and make it bright, despite certain limitations.

References

1. Lyuk.media. URL: https://lyuk.media/behind-city/hearing-loss/
2. Novak Y., Mazurets O. Practical Application of Method of Automated Personal Identification by Fingerprints Using Convolution Neural Networks. Proceedings of V International Scientific and Practical Conference «Modern strategies of global scientific solutions». December 27-29, 2023. Stockholm, Sweden, International Scientific Unity. 2023. Pp. 136-140.
3. Medium. Image Classification With MobileNet. URL: https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470
4. Stackoverflow. How does the flatten layer work in keras. URL: https://stackoverflow.com/questions/44176982/how-does-the-flatten-layer-work-in-keras
5. Wikipedia. Softmax. URL: https://uk.wikipedia.org/wiki/Softmax
6. GeeksforGeeks. Intuition of Adam Optimizer. URL: https://www.geeksforgeeks.org/intuition-of-adam-optimizer/#:~:text=Adam%20optimizer%20involves%20a%20combination,minima%20in%20a%20faster%20pace
7. Bohdanova A., Mazurets O., Sobko O. Gesture recognition using a neural network in real time. Black Sea Science 2023: Proceedings of the International Competition of Student Scientific Works. Odesa National University of Technology. Odesa, ONUT, 2023. Pp. 556-566.