# Research and development of image processing algorithms for effective recognition of various gestures in real time[1]*

Nataliya Boyko[1,†], Petro Telishevskyi[1,*,†] and Sotirios Argyroudis[2]

[1] *Lviv Polytechnic National University, Lviv 79013, Ukraine*
[2] *Brunel University London, London, United Kingdom*

**Abstract**
The study proposed a method of sign language communication using machine learning. Various sign language standards are considered. The study uses neural networks for gesture recognition, namely Convolutional Neural Networks (CNN). The work will also use OpenCV technology to capture gestures from video. The considered dataset for the neural network is ASL Alphabet. An overview of neural networks is provided, as well as a detailed description of how to apply and build a convolutional neural network. It is indicated by what means and where the development of the software product took place. The libraries that were used to perform the given task are described. The architecture of the convolutional neural network, which was used in the implementation of the software product, is indicated. After training, the neural network was tested and showed an accuracy of 90.16%. The software product is described and a user manual is created.

**Keywords**
computer vision, sign language, hearing impaired people, hearing impaired, neural networks, sign`s alphabet, convolutional neural networks

## 1. Introduction

When communicating, people with hearing impairments use sign language. Sign language is a type of speech that denotes individual words, phrases or letters to convey information. And for hearing-impaired people, mastering sign language is like mastering English or Ukrainian. And the problem is that people without hearing impairments in most cases do not know sign language and where does the communication barrier between people with hearing impairments arise. Because you need to use some methods to communicate with him, such as writing with a pen on paper to say what you want and others.

*SMARTINDUSTRY-2024: International Conference on Smart Automation & Robotics for Future Industry, April 18 - 20, 2024, Lviv, Ukraine*

* Corresponding author.
† These authors contributed equally.
nataliya.i.boyko@lpnu.ua (N. Boyko); ptelishewsky@gmail.com (P. Telishevskyi); sotirios.argyroudis@brunel.ac.uk (S. Argyroudis)
0000-0002-6962-9363 (N. Boyko); 0009-0008-8328-0373 (P. Telishevskyi); 0000-0002-8131-3038 (S. Argyroudis)

Nowadays, with advanced machine learning and computer vision, it is possible to solve the problem of the communication barrier. With the help of these technologies, it is possible to recognize and identify sign language gestures from a live video stream to convert into text or audio.

You may come across articles or studies that investigate this problem and the application of technology that solves this problem. Because according to statistics, 5% of people on Earth have hearing problems, and in numbers it will be approximately 446 million people, where 34 million are children. According to the WHO, the causes of hearing impairment are found in low- and middle-income countries.

As we already know, sign language is the main form of communication for people with hearing impairments. However, like ordinary language, there is no single standard for sign language in the world and there are 138 to 300 types. It can also be noted that there are three sign language standards for the international language ¬ English:

- American Sign Language (ASL);
- British Sign Language (BSL);
- Australian Sign Language (Auslan).

Therefore, there is an idea to solve the problem of the communication barrier with existing methods, where in the process they can be improved, or to develop your own method and present it in a form convenient for people. In our research, American Sign Language (ASL) will be the object of research. Let's break this down into some specific processes used to recognize sign language. This process steals the following items:

- We receive a video where the user wants to tell us in sign language;
- After receiving the video, we track the position of the hands on it and track them;
- With the help of machine learning, we determine which elements of sign language the user provides;
- In the form of a text field, as text information.

To simplify this process, the task of the master's qualification work will be to create software that, using machine learning methods, will simplify the recognition of sign language and translation into text. However, this task must be divided into several sub-tasks:

- Study of the subject area;
- Searching or creating data sets;
- Comparison of existing methods of problem solving;
- Analysis of the comparison result;
- Implementation of own solution based on comparison analysis;
- Analysis of finished results.

## 2. Review of the literature

The first post talks about how important sign language recognition is now because the hearing-impaired population is growing. It is also indicated that there is no trivial way to solve the problem, and for this, DCNN (Deep convolution neural networks) with transfer learning was used to solve the problem of the lack of a set of input data. Experimental results in cases using different data sets are described [1].

The second post covers the one-hand and two-hand Indian Sign Language recognition system. The author points out the shortcoming of the Zernike moments method for determining signs of hand gestures. The author suggests using methods of skin color segmentation to highlight signs of gestures and using the SVM (Support Vector Machine) method to classify sign languages [2].

In the third publication, it is said that for the recognition of sign language in dynamics, based on the approximation of the probability density function over a time interval, it is used. This is done to improve gesture classification. Results on datasets such as ASL, SHREC and LMDHG are also described [3].

The fourth publication described how random forest, decision tree and naïve Bayesian methods for gesture classification were studied for a sign language recognition system on two datasets, namely, ASLalphabet (American Sign Language) and ISL-HS (Irish Sign Language). In the system, random forest was chosen as the default method because the accuracy result was 96.7% on the ISL-HS dataset and 93.7% on the ASLalphabet dataset. The random forest method showed the best results [4, 10].

The fifth post suggests that languages use the TensorFlow Object Detection API to build a real-time sign language recognition system. Indian sign language recorded using a web camera was used to create the data set. The system showed quite good accuracy, which is 85.45% [5].

The sixth post shows how WSLR (Word-level sign language recognition) is used for sign language recognition. To improve spatial and temporal dependencies between different frames, it is suggested to use GCN (Graph Convolution Network) and BERT (Bidirectional Encoder Representations from Transformers). GCN should be used for better spatial interaction between videos, and BERT to fix the temporal dependence between frames [6, 8].

Reviewing scientific works during qualitative research, one can note the most common data sets for sign language recognition. Table 1 will indicate which datasets were used in the publications.

**Table 1**
List of common datasets in scientific papers

| The name of the data set | Hyperlink |
|---|---|
| How2Sign | https://how2sign.github.io/ |
| ASLalphabet | https://public.roboflow.com/object-detection/american-sign-language-letters |
| Indian Sign Language Dataset | https://www.kaggle.com/datasets/vaishnaviasonawane/indian-sign-language-dataset |
| ISL-HS | https://github.com/marlondcu/ISL |

## 3. Mathematical and algorithmic representation

### 3.1. General description of neural networks

In our work, an element of machine learning was used, to be specific, it is a neural network. To begin with, let's get acquainted with the basic meaning of a neural network, what it is, how it is built and what it consists of [13, 15].

Neural networks are the basis of deep learning algorithms. Neural networks are also called artificial neural networks. And the name is that they try to reproduce the activity of the human brain, namely to have structurally connected neurons that transmit signals to each other [7, 14].

We will give an example of how, in principle, the search for values in a neural network takes place, for this we will use the feedforward algorithm. Figure 1 will show an example of a neural network in which the value will be located.

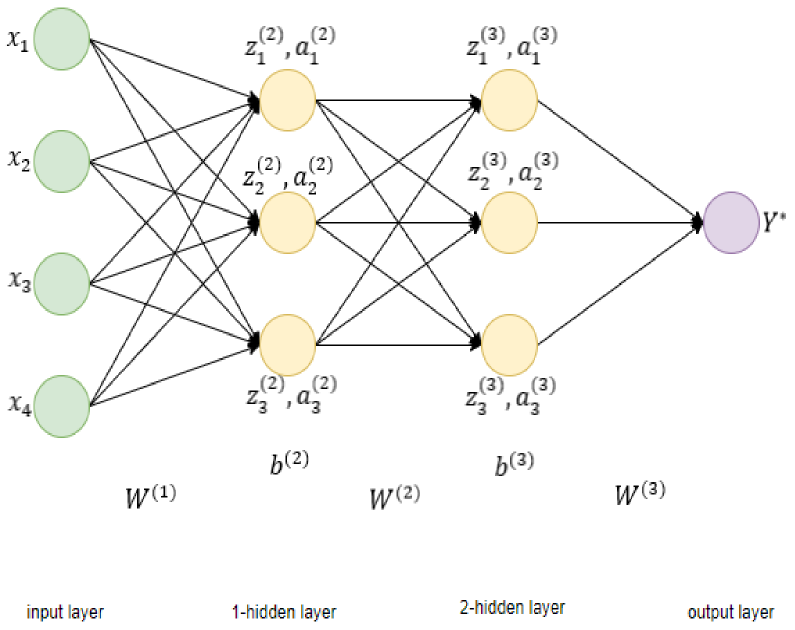


**Figure 1:** An example of a 4-layer neural network.

This network consists of four layers, namely: input, two hidden and output layers. We will need to find the following values of $z_i^{(l)}$, $a_i^{(l)}$ and Y*. To begin with, we apply the following formulas 1-2 to find $z_i^{(l)}$, $a_i^{(l)}$ [9, 11].

$$z^{(l)} = W^{(l)} \cdot a^{(l)} + b^{(l)} \tag{1}$$

$$a^{(l)} = f(z^{(l)}), \tag{2}$$

where *l* – the index of the hidden layer of the neural network;

*W* – weight matrix;

*b* – movement;

*a* – the activation function.

To calculate the predicted result Y*, we apply Formula 3.

$$Y^* = W^{(l)} \cdot a^{(l)}, \tag{3}$$

where *l* in our case will be equal to 3.

In the following subsections, we will consider what types of neural networks can be used for this study.

## 3.2. Convolutional neural networks

Convolutional neural networks or their other name in the English work – Convolution Neural Network (CNN), the abbreviation CNN will be used later in the work. This type of neural network will be used in our work [9, 12].

Convolutional neural networks are a deep learning algorithm that identifies certain features of the input image that make it possible to distinguish one from another. There are certain architectures of convolutional neural networks:

- LeNet-5;
- AlexNet
- VGGNet;
- GoogleNet;
- ResNet.

Let's consider how and from what a convolutional neural network is built and describe how it works. In a convolutional neural network, 3 main types of layers are used:

- Convolution layer (Convolution layer);
- Pooling layer (Unifying layer);
- Fully-connected layer.

Convolution layer is the foundation of convolutional neural network architecture. It is in this layer that features are extracted from the input image to be able to distinguish one image from another.

The convolutional layer has a filter or a so-called kernel in it, which makes it possible to extract features from the image. Typically, the most common kernel size is a 3 by 3 matrix, a 5 by 5 matrix, or a 7 by 7 matrix. On Figure 2 will show an example of how a kernel works in a convolutional layer [10].

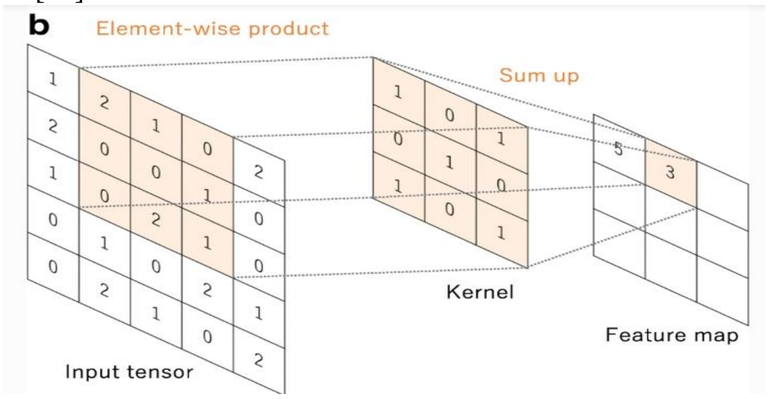


**Figure 2:** An example of a 3 by 3 kernel.

We will describe all parameters of the convolutional layer except for the kernel, namely [11, 16]:

- Number of filters;
- Stride;
- Padding.

Number of filters shows the output depth, namely the number of feature maps for the input image.

Stride is the distance or, as it is also called, the number of pixels that the kernel moves through the matrix of the input image.

Padding (indents) – this parameter is of four types:

- Zero-padding;
- Valid padding;
- Same padding;
- Full padding.

Zero-padding – if the kernel does not match the size of the matrix of the input image, then everything that stands for the size in the kernel is filled with zeros.

Valid padding – when this option is applied, padding is not applied and the last roll is not applied if the dimensions do not match.

Same padding – with this parameter, the output layer has the same size as the input layer.

Full padding – with this parameter, a frame of zeros is used so that the size of the output and input matches.

The pooling layer is responsible for dimensionality reduction by reducing the number of parameters in the input layer. As a result, this layer produces fewer features that can be studied. There are two types of pooling:

- Max pooling;
- Average pooling.

Max pooling – during this stage, the filter passing over the top of the matrix selects the pixel with the maximum value in the output array [12].

Average pooling – in this variant, the process is the same as in Max pooling. The difference is that here is the average value from the sample [13].

The fully-connected layer performs the task of classification based on features, namely from previous layers. The Fully-connected layer usually uses the softmax activation function to classify the data, producing a probability of 0 to 1.

## 3.3. How to activate the neuron

In our research, we have mentioned the activation function or transfer function. This element is important in neural networks. So, let's talk about it in more detail.

In order to know whether a neuron in a neural network needs to be fired or not, the network uses a transfer function. The mathematical formula of the activation function determines whether the neuron's input will affect the neural network's training process or not. Thus, it shows us the relationship between the output signal and the input.

There are different types of activation functions, but in our study, we will be interested in such a type as a nonlinear transfer function. Let us note that this type has many different functions. However, we will use ReLU, which in most cases is a transfer function for neural networks. According to formula 4, the ReLU function is performed, and Figure 3 shows the graph of the ReLU function [13, 15].

$$R(x) = \max(0, x) \tag{4}$$

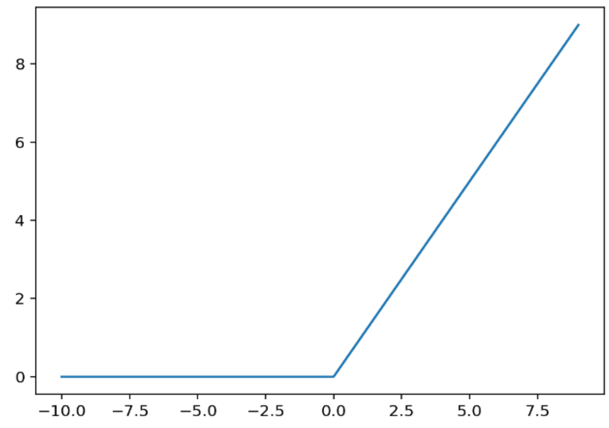


**Figure 3:** Graph of the ReLU function.

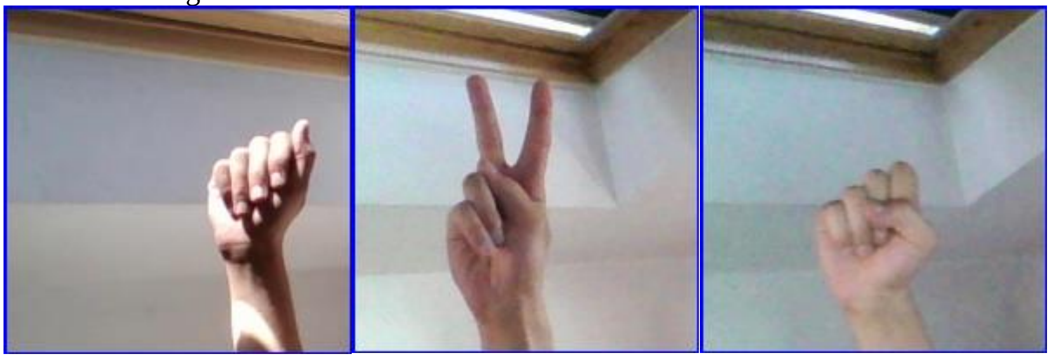
## 4. RESEARCH DEPARTMENT

### 4.1. Incoming data

In this research work, the object of research is signing language. If taken specifically, the work will consider the alphabet of gestures of American Sign Language. This alphabet will be in the form of images, where the gesture of each letter is shown on a separate image. Image data was taken from the ASL Alphabet dataset [14].

The ASL Alphabet dataset has images of the American Sign Language sign alphabet. The dataset contains 87,000 images of 200 by 200 pixels. The dataset includes 29 unique classes, where 26 classes are the letters A-Z and 3 classes for SPACE, DELETE, and NOTHING. These three classes (SPACE, DELETE and NOTHING) are needed to simplify classification in software products.

Let's also consider examples of alphabets that are in the ASL Alphabet dataset. Examples will be shown in Figure 4.



(a) (b) (c)

**Figure 4:** Example data from the ASL Alphabet dataset, where (a) is the letter A gesture, (b) is the letter K gesture, and (c) is the letter S gesture.

All these inputs will be used in training a neural network for gesture recognition. In the software implementation itself, the input data will be an image of a gesture recognized by a neural network.

## 4.2. Output data

What we can get as a result of our software implementation is text from gestures recognized by a neural network. That is, in our software application there is a streaming video where the user communicates in sign language. After that, a gesture is recognized on the video and a trained neural network classifies this gesture and it is broadcast in the form of text.

## 4.3. Data preparation

This subsection will describe how data was prepared for training a neural network that classifies user gestures. As already known in subsection 3.1, we use the ASL Alphabet dataset to train the neural network. As you already know, this set is built from folders where data is placed. However, this way we cannot provide data for training data, so we need to process it.

Let's give an example of how this happened for training data. To begin with, data was read from the training folder, where the names of the folders acted as a class for the data, and what was in the middle, namely the image, was read as data. Thus, we had two lists with images and classes, where the index of the image coincided with the index of the list of its class. During data mining, we resized the image from 200 by 200 pixels to 32 by 32 pixels.

Having two lists, we will prepare them for neural network training, namely, we will normalize the pixel values of the image to a common range. In this case, we used Formula 5.

$$normalizedPixel = \frac{pixel_{ij}}{255}, \tag{5}$$

where $pixel_{ij}$ is the value of a pixel in the image, *normalizedPixel* is the normalized pixel value.

We also convert the vector of classes into a binary matrix of classes. That is, we have classes from 0 to 28, where each number 0-25 is the value of the letters A-Z, and 26-28 is SPACING, DELETE, NOTHING. From where we convert 0-28 into a binary matrix of classes. How it will look will be depicted in Table 2, for example, let's take class 3, which means the letter D.

**Table 2**
Converting a class into a binary matrix of classes

| Class Letters | Binary representation of a class |
|---|---|
| 3(D) | [00010000000000000000000000000] |

The to_categorical() function from the keras library was used to convert a class into a binary matrix of classes.

## 5. Experiments

### 5.1. Neural network results

A convolutional neural network was used for the task of recognizing gestures of letters from sign language. This neural network is responsible for assigning a letter gesture to a specific class.

Consider the architecture of a convolutional neural network, Figure 5 will show the architecture itself.

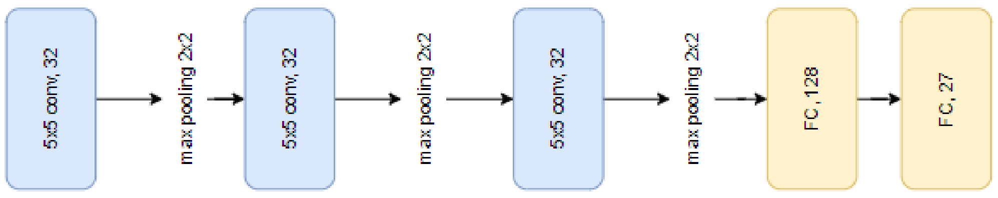


**Figure 5:** Architecture of a convolutional neural network.

From Figure 5, you can see that this neural network consists of five layers, where three of them are convolutional layers, and two of them are fully-connected layers. The optimizer used in neural network training is Adam.

The Adam optimizer is a stochastic gradient descent method that performs adaptive estimation of first and second order derivatives.

We will describe what parameters were used when training the neural network, this information will be entered in Table 3.

**Table 3**
Converting a class into a binary matrix of classes

| Parameter | Value |
|---|---|
| Epochs | 10 |
| Batch size | 128 |
| Learning rate | $10^{-4}$ |

After training the neural network, the network was tested, which resulted in 90.16% accuracy. Let's take a look at the discrepancy matrix to understand how the neural network model determines our letter gestures. Figure 6 shows this matrix.

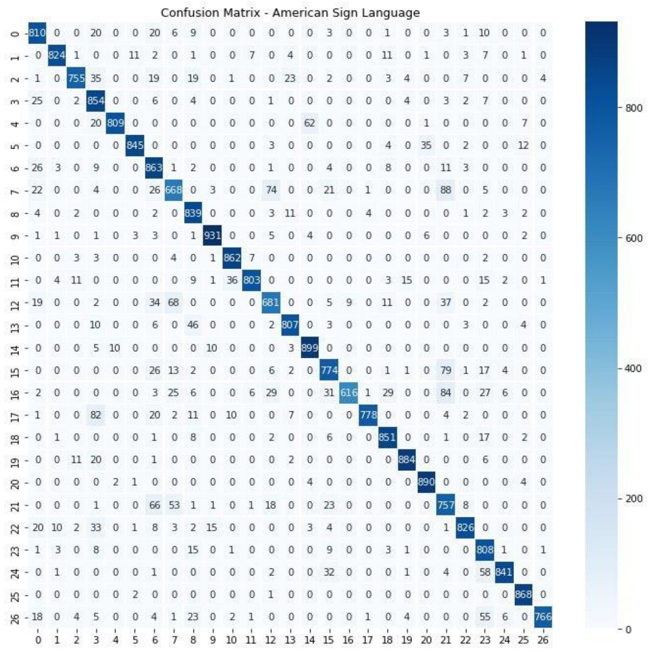
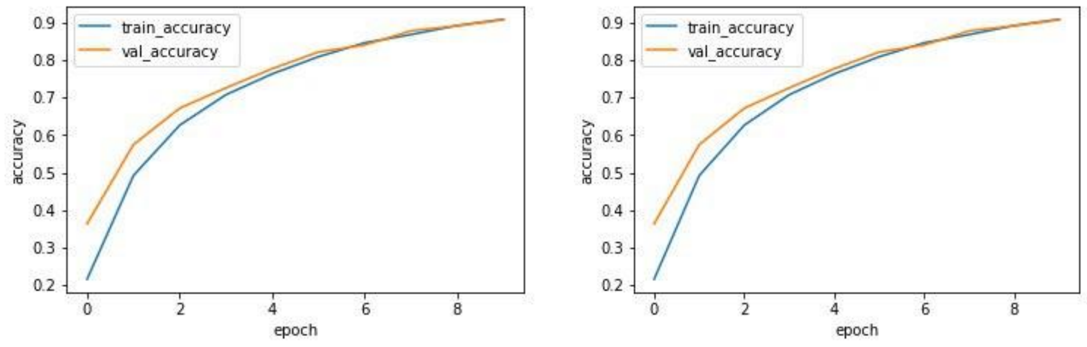


**Figure 6:** Confusion matrix.

We will also consider how the neural network behaved during training. Figures 7a, 7b will show graphs of accuracy and loss during neural network training.



a b

**Figure 7:** a) Neural network accuracy during training; b) Loss of a neural network during training.

When starting the software implementation, we will get the screen of this program, the screen will look like Figure 8.

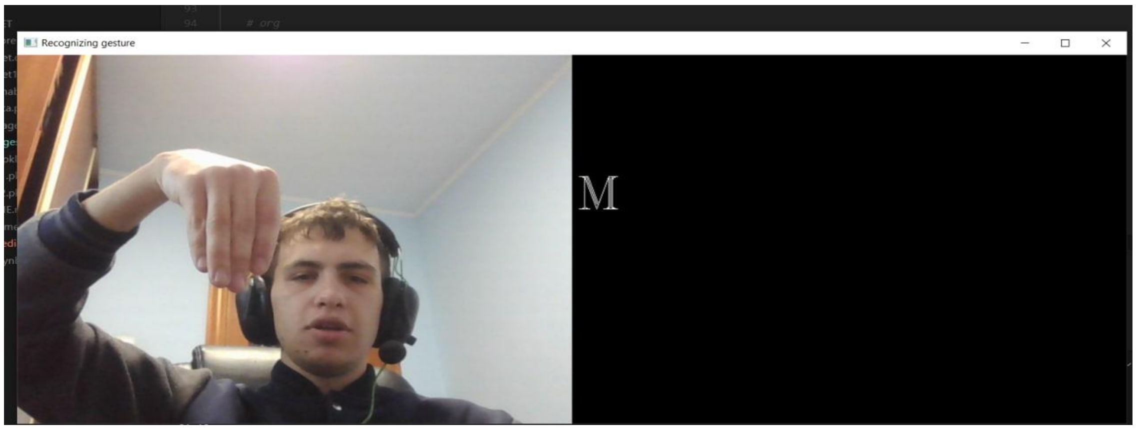


**Figure 8:** View of the screen of the software implementation.

From Figure 8, you can see that the screen of the software product is divided in half, where the video from the video camera is on the left, and the field where the text that was recognized from the gesture will appear is on the right. And the combination of gestures will form words and sentences.

Another important condition is that the hands or one hand is in the field of view of the web camera, as in Figure 8.

## 6. Conclusion

The software implementation of the software product was created using the technologies that were in operation. This study is relevant. Because every year there are more and more people with hearing impairments or hearing impairments, which creates social barriers with people without disabilities. And communication with these people will always be a problem, since sign language is not so easy to learn.

The implemented software product is an application that will make it possible to improve communication with people with hearing impairments. It can also be used as an integration into a large system, such as Zoom or Google Meet.

This topic and software implementation can be developed for quite a long time, because in sign language there are as many rules to be learned and applied that the software implementation will need to be improved.

## Acknowledgements

This research was funded by Ukrainian National Research Foundation, grant number 2021.01/0103.

## References

[1] Wang, Y., Wang, L., Zhao, Y. ”Research on Door Opening Operation of Mobile Robotic Arm Based on Reinforcement Learning.” Applied Sciences (2022): 12 (10), 5204. doi: 10.3390/app12105204.

[2] Jocher, G., et al.”Ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.” Zenodo (2022). doi: 10.5281/ZENODO.7347926.

[3] Arduengo, M., Torras, C., Sentis, L.”Robust and adaptive door operation with a mobile robot.” Intel Serv Robotics (2021): 14(3), 409–425. doi: 10.1007/s11370-021-00366-7.

[4] Huaman Quispe, A., Martinson, E., Oguchi, K.”Learning User Preferences for Robot-Human Handovers.” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2017). doi: 10.1109/IROS.2017.8202245

[5] Stuede, M., Nuelle, K., Tappe, S., Ortmaier, T.”Door opening and traversal with an industrial cartesian impedance controlled mobile robot.” International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada (2019): 966–972. doi: 10.1109/ICRA.2019.8793866.

[6] Quintana, B., Prieto, S. A., Adan, A., Bosche, F. ”Door detection in 3D colored laser scans for autonomous indoor navigation.” International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcala de Henares, Spain (2016): 1–8. doi: 10.1109/IPIN.2016.7743677.

[7] Papazov, C., Burschka, D.”An Efficient RANSAC for 3D Object Recognition in Noisy and Occluded Scenes.” Computer Vision - ACCV 2010 - 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part I. (2010). doi: 10.13140/2.1.1451.1041.

[8] Ahn, M. S., Chae, H., Noh, D., Nam, H., Hong, D. ”Analysis and Noise Modeling of the Intel RealSense D435 for Mobile Robots.” 16th International Conference on Ubiquitous Robots (UR), Jeju, Korea (South) (2019): 707–711. doi: 10.1109/URAI.2019.8768489.

[9] “Intel RealSense Documentation - Get Started,” Intel® RealSenseTM Developer Documentation. Accessed: Sep. 16, 2023. [Online]. Retrieved from: https://dev.intelrealsense.com/docs

[10] Hidalgo-Paniagua, A., Vega-Rodríguez, M. A., Pavón, N., Ferruz, J. (2015). A Comparative Study of Parallel RANSAC Implementations in 3D Space. Int J Parallel Prog. 43(5), 703–720. doi: 10.1007/s10766-014-0316-7.

[11] Chen, H., Liang, M., Liu, W., Wang, W., Liu, P. X. (2022). An approach to boundary detection for 3D point clouds based on DBSCAN clustering. Pattern Recognition. 124, 108431. doi: 10.1016/j.patcog.2021.108431.

[12] “Sklearn.cluster.DBSCAN,” scikit-learn. Accessed: Sep. 17, 2023. [Online]. Retrieved from: https://scikit-learn/stable/modules/generated/sklearn.cluster.DBSCAN.html

[13] “Depth Post-Processing for Intel® RealSenseTM Depth Camera D400 Series.” Accessed: Sep. 29, 2023. [Online]. Retrieved from: https://dev.intelrealsense.com/docs/depth-post-processing

[14] “Librealsense/examples/align at master · IntelRealSense/librealsense · GitHub.” Accessed: Sep. 29, 2023. [Online]. Retrieved from: https://github.com/IntelRealSense/librealsense/tree/master/examples/align#overview

[15] Chandra, S., Chrysos, G. G., Kokkinos, I. ”Surface Based Object Detection in RGBD Images.” Procedings of the British Machine Vision Conference 2015, Swansea: British Machine Vision Association (2015):187.1-187.13. doi: 10.5244/C.29.187.

[16] Zia, S., Yuksel, B., Yuret, D., Yemez, Y. ”RGB-D Object Recognition Using Deep Convolutional Neural Networks.” IEEE International Conference on Computer Vision Workshops (ICCVW) (2017): 887–894. doi: 10.1109/ICCVW.2017.109.