ISSN 2522-9818 (print)
ISSN 2524-2296 (online) *Innovative technologies and scientific solutions for industries. 2023. No. 4 (26)*

UDC 681.513.7; 519.688; 004.9 DOI: https://doi.org/10.30837/ITSSI.2023.26.108

Y. SHOVKOVYI, O. GRYNYOVA, S. UDOVENKO, L. CHALA

AUTOMATIC SIGN LANGUAGE TRANSLATION SYSTEM USING NEURAL NETWORK TECHNOLOGIES AND 3D ANIMATION

Implementation of automatic sign language translation software in the process of social inclusion of people with hearing impairment is an important task. Social inclusion for people with hearing disabilities is an acute problem that must be solved in the context of the development of IT technologies and legislative initiatives that ensure the rights of people with disabilities and their equal opportunities. This substantiates the relevance of the research of assistive technologies, in the context of software tools, such as the process of social inclusion of people with severe hearing impairment in society. **The subject of research** is methods of automated sign language translation using intelligent technologies. **The purpose of the work** is the development and research of sign language automation methods to improve the quality of life of people with hearing impairments in accordance with the "Goals of Sustainable Development of Ukraine" (in the "Reduction of Inequality" part). **The main tasks** of the research are the development and testing of methods of converting sign language into text, converting text into sign language, as well as automating translation from one sign language to another sign language using modern intelligent technologies. Neural network modeling and 3D animation methods were used to solve these problems. The following **results** were obtained in the work: the main problems and tasks of social inclusion for people with hearing impairments were identified; a comparative analysis of modern methods and software platforms of automatic sign language translation was carried out; a system combining the SL-to-Text method is proposed and investigated; the Text-to-SL method using 3D animation to generate sign language concepts; the method of generating a 3D-animated gesture from video recordings; method of implementing the Sign Language1 to Sign Language2 technology. For gesture recognition, a convolutional neural network model is used, which is trained using imported and system-generated datasets of video gestures. The trained model has a high recognition accuracy (98.52%). The creation of a 3D model for displaying the gesture on the screen and its processing took place in the Unity 3D environment. The structure of the project, executive and auxiliary files used to build 3D animation for the generation of sign language concepts includes: event handler files; display results according to which they carry information about the position of the tracked points of the body; files that store the characteristics of materials that have been added to certain body mapping points. **Conclusions:** the proposed methods of automated translation have practical significance, which is confirmed by the demo versions of the software applications "Sign Language to Text" and "Text to Sign Language". A promising direction for continuing research on the topic of the work is the improvement of SL1-to-SL2 methods, the creation of open datasets of video gestures, the joining of scientists and developers to fill dictionaries with concepts of various sign languages.

Keywords: automation of sign speech; animated character; body position tracking; people with hearing impairments; sign language; neural networks; gesture recognition; ukrainian sign language; sign language translation; reduce inequality.

Introduction

Sign language is one of the oldest ways of communication for people with hearing impairments in all cultures, with its own rules and norms. However, despite the fact that sign language is no less important than verbal language, it has not received sufficient attention in research. This language consists of gestures, each of which is performed with the hands in combination with body position, facial expressions, shape or movement of the mouth and lips. The use of sign languages by people without hearing loss is secondary, but quite common, because there is often a need to communicate with people with hearing loss who use sign language. The use of sign language instead of voice communication can also be useful in situations where voice communication is impossible or difficult.

According to the World Health Organization (WHO), approximately 1.5 billion people (approximately 20% of the population) worldwide have hearing impairments, of which 430 million are completely deaf (approximately 5.7% of the population) [1]. More than 44 thousand people with hearing impairments are registered with the All-Ukrainian public organization Ukrainian Society of the Deaf. These citizens need attention and protection because of their hearing impairment, especially during the military aggression on the territory of Ukraine.

Today, many web developers are trying to make their sites more accessible to people with hearing impairments. For this purpose, in particular, WCAG web accessibility standards are used [2]. It should be noted that in most developed countries there is legislation that obliges organizations and companies to provide accessibility to their resources for users with various disabilities, including people with hearing impairments. The use of special software tools allows this category of people to develop the practical skills and abilities necessary for education, communication with the outside world, assimilation of information, further employment, building a professional career and living in general. The Decree of the President of Ukraine "On the Sustainable Development Goals of Ukraine for the period up to 2030" (in the part "Reducing Inequality") recommends taking this aspect into account when determining the directions of scientific research [3].

Thus, social inclusion for people with hearing impairments is an acute problem that needs to be addressed in the context of the development of IT technologies and legislative initiatives that ensure the rights and equal opportunities of people with disabilities. This justifies the need and relevance of researching assistive technologies and software tools to facilitate the social inclusion of people with severe hearing impairments in society. Such impairments are partially compensated for by creating appropriate living conditions, which are achieved with the help of modern information technologies. The problem of using sign language is that this type of communication is extremely important for people, especially for those who have limitations in verbal communication, such as people with autism or hearing impairments. In addition, the growth of globalization and intercultural communication has made sign language even more relevant, as it can influence the perception of cultural differences and promote mutual understanding between people from different cultures and language groups. The study of sign language can open up new opportunities for those who have limitations in verbal communication and help to understand which gestures are acceptable for different cultures and situations.

Ukrainian Sign Language (USL) is a natural language system that is transmitted by visual and gestural means and has its own lexical and grammatical structure [4]. It has naturally developed and serves as a primary mode of communication for individuals using sign language residing in Ukraine, whether currently or in the past. In the 20th century, the USL began to develop actively and become more standardized. In 1963, a scientific laboratory of sign language was established at the Taras Shevchenko National University of Kyiv. As of January 1, 2022, there were about 40 preschools in Ukraine with deaf children under 6 years of age and 60 specialized general education schools for deaf students between the ages of 6 and 18. From 2006 to 2018, the regulatory authority in Ukraine that was responsible for the study of linguistic features and the development of DMC published textbooks, manuals and scientific articles. The COVID-19 pandemic had a significant impact on the communication of people with hearing impairments around the world. As most people with hearing impairments depend on lip reading and visual communication, the introduction of restrictions and recommendations on social distancing and the use of masks became significant barriers to communication. Since February 24, 2022, the hostilities have particularly exacerbated the isolation and alienation of deaf Ukrainians and increased the threat to their lives. Ukrainians with hearing impairments faced a number of problems during the martial law: first, they do not have access to information about air alerts, as they cannot hear sirens or telephone signals; the second problem was the lack of access to the local administration's hotline, rescue service, remote consultations of medical and psychological workers, volunteers, etc.

Raising awareness and understanding among the general public of the problems faced by people with hearing impairments should help reduce discrimination and improve attitudes towards this category of people. To solve these problems, experts and specialists in the field of sign language interpretation, linguistics, artificial intelligence, and software development should be involved to comprehensively address the problems described above. It is necessary to offer accessible software applications for easy use in everyday life, as this will help to ensure the rights of people with hearing impairments and their participation in society. Therefore, the study of models and methods of automated sign language translation (namely, translation of sign language into text or sounds and vice versa, as well as translation from one sign language to another sign language) is an urgent task.

Comparative analysis of existing sign language interpretation systems and technologies. Purpose and objectives of the study

Modern technologies that are used to communicate between people with hearing impairments can be classified according to the following features:

a) the possibility to choose a sign language (SL): one or more;

b) the type of translation:

– Speech → Text;

– SL → Text;
– SL → Text → Speech;
– Text → SL;
– Speech → Text → SL;
– Text → Speech;
– Text1 → Text2;
– Speech1 → Speech2;
– SL1 → SL2;

c) the object of translation: a person or a program.

Speech → Text is an important type of communication for people with hearing impairments. An example of this is the use of subtitles, which can be useful for a variety of reasons: to help people with hearing loss who cannot clearly hear or understand the language used in a video or movie; to improve language comprehension (particularly in foreign language learning); to improve listening (subtitles can help people improve their listening skills by allowing them to listen more closely and understand content); to allow content to be translated into other languages, allowing people from different countries and cultures to understand the content.

Speech → SL is also a common type of communication from a hearing person to a person with a hearing impairment. This method describes the work of sign language interpreters, whose main tasks include: real-time translation of direct speech for hearing impaired people who interact with the world using SL; preparation for important events (conferences, meetings and other events where hearing impaired people may be present); creating subtitles for videos and movies to make them more accessible individuals with hearing loss; preparation of educational materials for hearing impaired people to provide them access to education. Sign language interpretation systems are used in various industries, including education, medicine, government agencies, etc. In particular, they use services such as Google Meet, Zoom, Skype Translator, WebEx, Microsoft teams, as well as applications for mobile devices (Ava, iTranslate, Sorenson Buzz). Some of these services provide free access to sign language interpretation, while others may be paid. A common mobile application from Microsoft is the Seeing AI app, which uses machine learning and artificial intelligence to help people with visual impairments. The program is available for iOS and Android and can be downloaded for free from the App Store or Google Play. One of the main features of Seeing AI is sign language recognition, which allows the user to translate gestures into text in several languages, including Ukrainian. To do this, you need to point the camera of your mobile device at the person using the app, and the program will automatically recognize gestures and translate them into the text for chosen language.

Let's analyze the programs and technologies that allow you to translate text into SL (Text → SL). Some of them are used for online translation, while others are designed for use in educational institutions and rehabilitation centers. The most popular programs for translating text into SL include Signing Savvy, VL2 Signing Avatar, Spread The Sign, ProDeaf, and Hand Talk. Signing Savvy's online sign language dictionary allows users to look up and translate words and phrases into sign language, and provides videos demonstrating how to perform gestures and options for adjusting the video recording speed.

Let's take a look at some programs and technologies that use 3D animation to translate text into sign language. These programs typically use computer vision and machine learning technologies to create animated characters that can perform gestures and convey messages into sign language. The most popular programs of this type include SigningAvatar, JASigning, and Signily. SigningAvatar is a system that automatically translates text into SL using 3D animation. This technology allows users to enter text in natural language and receive videos with translation in the GUI. The main advantages of SigningAvatar are automatic translation, high accuracy, multilingualism, and ease of use. Disadvantages of SigningAvatar: American English-language version of the SL; limited set of gestures; limited user options (SigningAvatar does not allow users to create their own gestures or edit existing gestures).

Currently, there are several software-based sign language interpretation systems in Ukrainian that can be useful for people with hearing impairments (in particular, SignTalk, Ava, and SignAll). SignTalk is a software-based sign language interpreting system that uses SL to transmit messages between hearing impaired users and non-HSL users. SignTalk has a Ukrainian-language interface and can be used on computers and mobile devices. Ava is an application for mobile devices that uses machine learning to translate speech into text in real time and supports several languages, including Ukrainian. SignAll is a software system that uses modern gesture recognition technologies, including deep learning, to accurately translate speech into Ukrainian text in real time. The system uses three video cameras and one graphics card to collect and process information about hand movements and gestures.

It should be noted that mentioned systems, which are used to communicate between individuals with hearing loss, are constantly being improved using various modifications of artificial neural networks and 3D animation tools.

Let us consider some results of modern research concerning the practical implementation of communication methods for people with hearing impairments and confirming the relevance of the problem of creating software tools for automating sign language interpretation.

Article [5] presents an overview of some methods of segmentation techniques for recognizing hand gestures. This study is aimed at using hand gesture recognition for sign language interpretation in the process of human-computer interaction. Segmentation uses various hand detection schemes with the necessary morphological processing. Paper [6] describes a system capable of efficiently converting sign language gestures into spoken language. This system aims to provide voice output in various regional languages. In article [7], proposed model extracts temporal and spatial characteristics after capturing a video sequence. To extract the spatial characteristics, a program was used to determine the landmarks of the face, face, and pose using various types of RNNs (recurrent neural networks), including LSTM and GRU. Paper [8] proposes a variant of the perceptual computing user interface that allows computers to capture and interpret human gestures as commands. Paper [9] proposes a system for recognizing hand gesture images using modern image processing methods. During image segmentation, skin color detection and morphological operations are performed to accurately segment a part of the hand gestures. Then, a heuristic manta ray feature optimization (HMFO) technique is used to optimally select features by calculating the best fitness value. Paper [10] proposes a method for detecting 3D objects based on a point cloud and a graphical neural network (GNN) in combination with an attention mechanism. The paper [11] proposes a spatio-temporal GCN model for adaptive construction of spatio-temporal graphs that allow creating sign language recognition datasets based on a video skeleton. Article [12] provides an overview of research based on the use of hybrid recurrent neural networks (RNNs) to build a gesture recognition system that can reduce the number of classification errors and increase the stability of recognition. Article [13] provides an overview of research based on the use of hybrid recurrent neural networks (RNNs) to build a gesture recognition system that can reduce the number of classification errors and increase the stability of recognition. In article [14], a convolutional neural network model for hand gesture recognition is proposed. In this paper, a one-time coding technique is used to convert categorical data values into binary form. Unimportant parameters are excluded from consideration, which improves classification accuracy. The paper [15] proposes the idea of static and dynamic capture and recognition of human gestures in real time based on a radial basis function neural network (RBFNN). Dynamic time warping (DTW) is used to select candidates for dynamic behavior, as well as to recognize gestures by comparing observed recordings with a series of pre-recorded reference data templates. In article [16], a segmentation method is proposed to identify hand gestures from an input image, which improves recognition accuracy. A comparison of hue and saturation segmentation methods for different background lighting conditions is presented. In article [17], proposed model can recognize hand gestures and signs using a convolutional neural network (CNN) and convert them to text.

Of course, the programs and technologies discussed are still evolving, and each has its own advantages and limitations. Nevertheless, they are an important step in providing access to sign language for people with hearing impairments and help make sign language more accessible to people without hearing impairments. Based on the results of the analysis of scientific publications and modern technologies devoted to the automation of sign language interpretation by software tools, we formulate the purpose and objectives of the proposed work.

The aim of the work is to develop and study methods of sign language interpretation automation that improve the level of communication for people with hearing impairments in accordance with the Sustainable Development Goals of Ukraine.

Research objectives:

– developing methods for converting sign language into text, converting text into sign language, and automating translation from one sign language to another sign language using modern intelligent technologies;

– testing of the proposed methods of sign language translation automation and determination of prospects for their application.

Proposed technology for automatic sign language interpretation

The technology proposed in this paper involves the implementation of methods for converting sign language into text (task A) and methods for converting text into sign language (task B) using neural networks and 3D animation.

Let's first consider the essence of the implementation of task A. Sign language to text (SL-to-Text) technologies are a set of various methods that can be used to convert gestures used in sign language into written text. The main goal of SL-to-Text methods is to help people with communication disabilities use the web as a means of interacting with the world around them by converting their gestures into text phrases. The proposed SL-to-Text method can be divided into three stages.

The first stage involves capturing the image of a gesture (or movement) of all parts of the body using video devices and software. At this stage, it is necessary to capture a video object using a webcam or other video device to obtain an image of a hand performing a gesture against the body background. Specialized software is used to capture and further process the image to be analyzed.

The second stage involves determining the contours of the hand and its location in space in the image using computer vision algorithms. In particular, to recognize the hand as an object, a color filter and markup algorithm is used. Skin color detection is achieved by detecting a skin area where skin pixels are spatially processed based on intensity and texture analysis. The resulting black-and-white image allows for the identification of the palm (in Fig. 1, its image is white). Then the palm is framed by a rectangular outline, which further helps to track the position of the hand (Fig. 1).

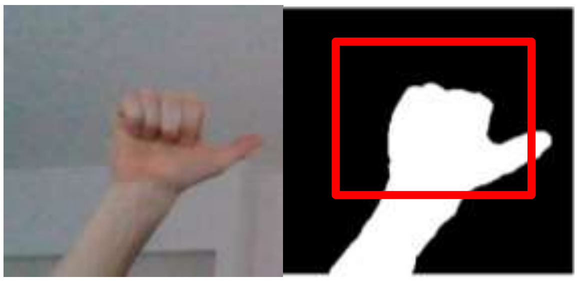

Fig. 1. An example of capturing the "Palm" object with a webcam

The third stage involves gesture recognition. Once the hand is detected in the image, the system identifies the intended gesture. For gesture recognition, a convolutional neural network (CNN) model is used, which is trained on imported and self-generated video gesture datasets. In general, a CNN is an artificial neural network that specializes in the ability to select or detect and interpret certain patterns. The use of existing patterns allows the CNN network to analyze frames of gesture images. The CNN consists of: hidden convolutional layers with filters capable of detecting patterns and a nonlinear function (usually ReLu); pooling layers to reduce the number of parameters for too complex images; a fully-connected layer that transforms the combined image matrix into a vector and then applies the Softmax function to classify the object (Table 1).

Table 1. *CNN architecture model*

Stages / Operations	**Formal models of operations implementation**
Convolution	$z^{l} = h^{l-1} * W^{l}$ (1)
Pooling, Max-Pooling	$h^{l}_{xy} = \max_{i=0,...,s,\ j=0,...,s} h^{l-1}_{(x+i)(y+j)}$ (2)
Fully-connected layer	$z_{l} = W_{l} * h_{l-1}$ (3)
ReLu(Rectifier)	$\mathrm{ReLU}(z_{i}) = \max(0, z_{i})$ (4)
Softmax	$\mathrm{softmax}(z_{i}) = e^{z_i} \Big/ \sum_{j} e^{z_j}$ (5)

The following notations are used in Table 1:

z^{l} – output of convolution layer l;

h^{l} – activation of layer l;

$*$ – discrete convolution operator;

W – training parameter.

The built model is tested on a pre-created test sample of gesture images obtained under different lighting conditions, on different backgrounds, etc. The result of using the model is text classified by gesture analysis. To ensure better gesture translation, it is advisable to add such operations as noise filtering, image alignment, adaptation to different lighting and camera movement to the basic SL-to-Text method procedure. In general, the above approach for implementing SL-to-Text using the CNN architecture is universal for solving gesture recognition tasks and processing corresponding images. However, it is important to pay attention to the specifics of a particular task and the context in which it is performed. An example of gesture classification using the Sign Language to Text method is shown in Fig. 2.

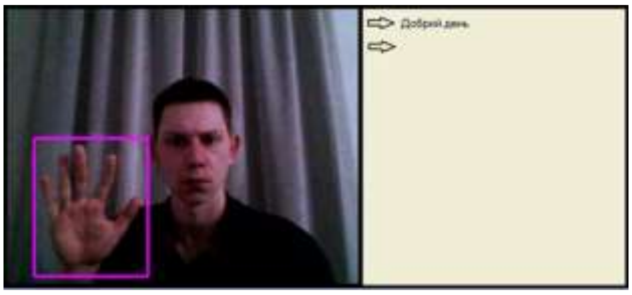

Fig. 2. An example of gesture classification using the Sign Language to Text method

Let us further consider the essence of text-to-sign language conversion according to the proposed approach (Text-to-SL and Speech-to-SL), which uses computer vision and machine learning methods to convert written or spoken text into SL. It should be noted that each SL has its own rules and grammar that differ significantly from the rules of written language, so it is impossible to make a direct word-for-word translation, but it is possible to convey the very idea of the statement, its keywords or phrases.

At the first stage of implementing the Text-to-SL method, the initial input text is obtained, which needs to be translated into a SL while preserving the idea behind the text. There are two ways to obtain the initial input text: entering text using the keyboard (Text to Text Summarization) or recording voice through a microphone and then recognizing speech, the sounds of which are converted into text (Speech-to-Text stage).

In the case of Speech-to-Text, the received audio file must be pre-processed, which involves noise reduction, filtering, and equalization. Next, various acoustic features are extracted from the audio signal, such as frequency, energy, sound pressure, etc. Based on the extracted acoustic features, speech is decoded using a hidden Markov model, which is widely used for pattern recognition in speech. The phoneme is divided into smaller sound elements, each of which corresponds to a state:

$$a_{ij} = P\left[q_{t+1} = S_j \mid q_t = S_i\right], \; a_{ij} \le 0, \; \sum_{j=1}^{N} a_{ij} = 1,$$
$$B = \left\{b_j(k)\right\}, \; b_j(k) = P\left[v_k \; at \; t \mid q_t = S_j\right], \qquad (6)$$
$$\pi_i = P\left[q_i = S_i\right], \; 1 \le i, j \le N, \; 1 \le k \le M,$$

where N is the number of states of the model;

S_j is the state of the model (at time t denoted as q_t);

M is the size of the discrete alphabet;

A is the probability transition matrix $\left(A = \left\{a_{ij}\right\}\right)$;

B is the probability distribution of the appearance of observation symbols in state j $\left(B = \left\{b_j(k)\right\}\right)$;

vk is the observed symbol;

π is the initial probability distribution of states $\left(\pi = \left\{\pi_i\right\}\right)$.

After choosing the values for N, M, A, B, and π, the neural network model can be used to generate a sequence of observations $O = O_1 O_2 ... O_L$, where each observation O_t is a character from the alphabet V, and L is the number of characters in the observed sequence. Thus, by training the model on test recordings, it is possible to obtain the primary text T_0 corresponding to certain sounds and use it in subsequent stages to generate the corresponding gestures. In the case of Text to Text Summarization, it is necessary to perform a semantic analysis of the primary text T_0 and generate a new reduced text T_1, i.e., determine the meaning to be conveyed through gestures. To do this, we use Text Summarization methods that allow us to recognize language concepts based on context, phrase and phrase analysis. After the semantic analysis of the T_1 text, it is necessary to select gestures or sets of gestures that best match the semantics of the T_1 text concepts. Rule-based machine translation (RBMT) and the use of a dictionary of SL concepts allow for the selection of SL concepts. These concepts include the use of SL signs, gestures that reflect movement and space, and gestures that convey emotions and mood. That is, the result of this stage should be a word or phrase, the meaning of which can be further conveyed through gestures.

To solve this problem, it is proposed to use SL dictionaries, in particular, ASL Pro Dictionary, Lifeprint, Signing Saavy, etc.

The selection of gesture concepts is carried out according to an algorithm that involves the sequential implementation of points P1 – P7:

– P1. The input sentence is divided into fragments (words, phrases);

– P2. The selected fragments are searched in the available SL dictionaries;

– P3. If the fragment is present in the SL dictionaries, you go to P5;

– P4. If the fragment is present in the SL dictionaries, you go to P6;

– P5. The selected video fragment is saved in the current buffer of gesture concepts for further visualization (go to P7);

– P6. The fragment is excluded from the list of candidates for inclusion in the set of gesture concepts, and its SL dactylic translation is used to further form sets of gesture sequences that convey the semantics of the sentence, followed by a transition to P7;

– P7. A set of video gesture concepts is formed (according to P5) and a set of fingerprint displays of individual sentence fragments.

Dactylic translation of text into sign language implies the need to divide the analyzed sentence fragment into letters and symbols, followed by displaying the corresponding images of the position of the hand elements at intervals of several seconds.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2023. No. 4 (26)

It should be noted that the presence of a large number of fingerprint mappings (as a percentage) compared to the number of video gesture concepts is undesirable, as it can significantly increase the duration of real-time sign language interpretation. However, fingerprint mappings allow for a fairly accurate representation of some specific information (e.g., contact information and proper names of interlocutors).

After the SL concepts are selected (according to this algorithm), sets of gesture sequences that convey the semantics of the T1 text are generated (they should include determining the order of gestures, taking into account the tempo and intensity, and using gestures that emphasize important information). Once a sequence of gesture concepts has been formed, it is necessary to graphically visualize these gesture sets on device screens so that users can see them (the SL gesture concept visualization stage). You can display a demonstration of a pre-recorded video gesture or an animated character. There are quite large databases of recorded video gestures for various SLs, as well as recommendations for the formation of a library of animated SL concepts (development of 3D characters by animation programmers based on a video sample).

In this paper, we propose to automate the process of filling the database of animated gesture concepts by generating an animated gesture (AG) by an AI model from a video file recorded by a person and saving it in a database (Fig. 3). It is necessary to develop a neural network model that will receive a video file as input and a csv (comma separated values) file as output, where each line contains the time and a set of hand points in space for each frame. Next, you need to bind each point to the corresponding point of the 3D character and change their position in space with each frame. The output is an AG with a description that is stored in the database of all animated gestures.

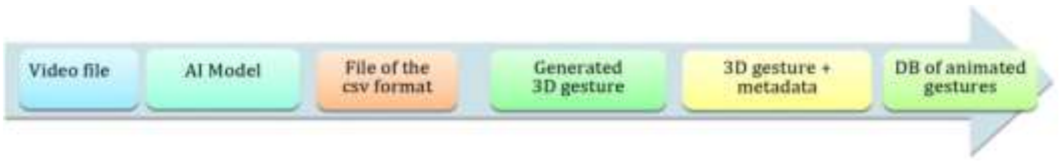

Fig. 3. Scheme of animated gesture generation

Using a dataset of video files with gesture demonstrations, the AI model was trained to generate a 3D gesture as a set of connected labeled points. The position of the hands (including the palm) in space during the gesture was tracked and recorded frame by frame. To do this, we used the method of Dung and Mizukawa, which implements a technique that can be used to extract the center of the hand and a group of feature pixels called distance feature pixels. For these pixels, the Hough transform is computed to detect all the extended fingers as lines. Once the lines are detected, the directions and positions of the fingers can be accurately determined. With a set of points previously saved in a csv file that correspond to the position of the hand in each frame, you can animate a 3D character and save the generated animated gesture to the database. The result of this step is a sequence of short animated hand gestures of a 3D character found in the database. At the same time, we get simplified phrases in SL according to the semantics of the T0 text that needed to be translated. An example of using the created software application "Text to Sign Language" (to translate a phrase into SL) is shown in Fig. 4. The trained AI model can be improved in such a way that 3D gestures are generated during communication, rather than from a database of animated gestures.

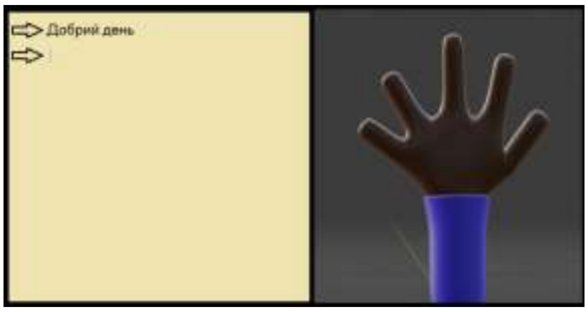

Fig. 4. An example of translating a phrase into SL using the "Text to Sign Language" method

The relevance of developing the Sign Language1 to Sign Language2 method is due to the problem of communication between people who speak different sign languages. Let us consider the features of the proposed variant of this method, the idea of which is to combine the technologies described above in order to obtain a general scheme of translation from one sign language to another:

– the interlocutor selects the SL1 sign language he or she speaks on the screen (Ukrainian, English, Spanish, German, etc.);

– the next step is to record a video of the interlocutor's gesture on the camera (the camera should be positioned so that the upper part of the body above the waist is visible);

– the video is then processed by the program, which results in the text Text1 appearing on the screen, which the program recognized from the gesture;

– the other person chooses the SL2 language in which he or she is communicating;

– Text1 is translated into SL2, resulting in the translated Text2;

– from Text2, the meaning of which must be understood by the other interlocutor in SL2, the corresponding gestures are generated and demonstrated on the screen using a 3D animated character in SL2.

An example of the implementation of this method for communication between hearing impaired users speaking different SLs is shown in Fig. 5.

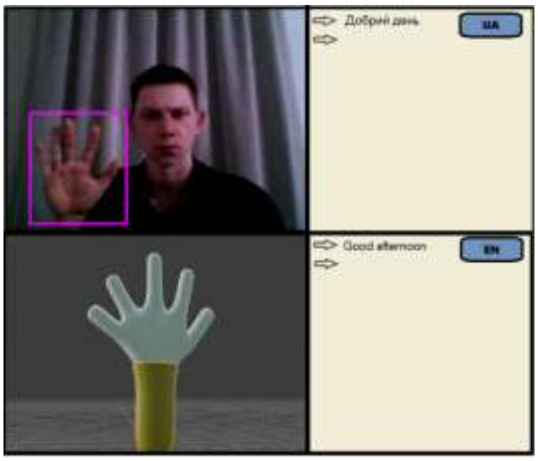

Fig. 5. An example of sign language translation using the "Sign Language1 to Sign Language2" method

Such a software application can be especially relevant if a hearing impaired user travels to another country but does not know the specifics of its SL. With the proposed software application, the user will be able to use the camera of their device to recognize gestures and receive appropriate translations or interpretations of gestures in the language of the host country. This software application can also be useful for those who need translation or interpretation of gestures in various life situations (in particular, for communication between users with and without hearing impairments during a business trip).

Software implementation and testing results of the proposed automatic sign language interpretation technology

Let us justify the choice of programming language, libraries and development environment for the implementation of the Sign Language to Text and Text to Sign Language methods.

To solve the problem (SL-to-Text), we used the open source programming language Python. One of the advantages of this language is the availability of a wide range of libraries (including those for working with neural networks), tools, and frameworks that facilitate the process of quickly creating complex programs. PyCharm integrated development environment (IDE) was chosen to implement the Sign Language to Text method. This environment offers an extensive list of features and tools for Python developers, including code highlighting, code navigation, testing, and version control integration.

To implement the tasks of tracking the position of the hands and body in the proposed technology, the open source Mediapipe platform from Google was used. One of the main advantages of this platform is that it provides pre-built and optimized algorithms and models that can be easily integrated into applications, and offers a variety of tools that allow developers to create complex computer vision and machine learning applications.

Intel's OpenCV computer vision and machine learning library, which provides interfaces to Python, was used to render and display the contours of certain body parts found with Mediapipe. OpenCV contains a number of optimization algorithms that can be used to perform image and video processing tasks such as object detection, recognition, tracking, segmentation, and more.

TensorFlow and Keras libraries were used to build a neural network model for recognizing gestures from the screen. TensorFlow is an open source platform from Google that provides a wide range of built-in operations and functions for creating and training neural networks. One of the key advantages of the TensorFlow library is the ability to take advantage of GPUs to accelerate training. Keras is an open-source deep learning library used as an API for building and training deep neural networks. Keras is based on TensorFlow and provides a user-friendly interface for building and training models, which makes it easy to experiment with different architectures and hyperparameters. Keras also provides a variety of pre-built layers and models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which makes it easy to develop deep learning models. Keras also includes various tools for model visualization, as well as tools for data preparation and preprocessing.

To create a 3D model of the gesture demonstration, we used the 3D Unity engine, which allows us to support 3D graphics, as well as visualize the architecture and model various types of interactive media [10].

Let us consider the structure of projects for the software implementation of the Sign Language to Text and Text to Sign Language methods using the listed development environments, libraries, and programming languages.

The Sign Language to Text method and several stages of the Text to Sign Language method were implemented in Python in the PyCharm development environment using the libraries described above. The structure of the PyCharm project consists of the following directories and executive files:

– Dataset (the directory contains a set of points in space corresponding to each gesture on which the deep neural network model was trained);

– Logs (the directory contains log files during model training that allow visualizing the model training process and obtaining the desired statistics from certain stages of the process);

– main.py (this file contains the implementation of the Sign Language to Text method at all its stages: using a camera to record gestures, libraries to track body position, storing body position points in space, preparing, creating and training the neural network model itself, and displaying the results on the screen);

– sl_to_text.h5 (trained neural network model for further use when running the program);

– text-to-sl-csv.py (a file containing the implementation of the stage of collecting points in space to build a 3D model of the Text to Sign Language method);

– video (a directory containing the video files for the text-to-sl-csv.py file, from which the body position points for the 3D model are tracked).

The 3D model for displaying the gesture on the screen was created and processed in the Unity 3D development environment. The structure of this project consists of the event handler files BodyAnimation.cs, LeftHandAnimation.cs, RightHandAnimation.cs; the files for processing the display result for each frame LineCode.cs; the files pose.txt, lh.txt, rh.txt, which contain information about the position of the controlled body points; the files Blue.mat, Green.mat, Red.mat, which store the characteristics that were added to certain body display points. The project structure also includes the Scenes directory, which contains the system files for setting up the scene, as well as the pose_2test.txt, lh_2test.txt, rh_2test.txt files with the body position points of additional gestures that were added to test the developed program.

The Sign Language to Text method is implemented in 6 stages. Below is their description.

The first stage (importing libraries). To implement this stage, the PyCharm development environment uses a developed library import program.

Among the imported libraries are libraries for working with OpenCV, numpy data arrays, the os operating system, matplotlib graphical visualization, time, and the keras library for working with neural networks.

The second stage (using OpenCV to connect the camera and display it on the screen). First, you need to specify which camera will be used in the process (built-in or from an external device). Then, all the collected information from the camera is transferred to the loop, which changes the image on the screen with each frame. The camera can be closed by pressing the 'Q' key or simply by closing the camera window on the monitor screen. The program code for this stage was developed

The third stage (using the Mediapipe platform to track the position of body parts).

To implement this stage, we used the Mediapipe library module for recognizing the necessary body points and the module for displaying it on the screen. Mediapipe makes it possible to track hands, body posture, face, and all of the above separately.

In addition, to track the body with the desired parameters, a function was developed and programmatically implemented, the input parameters of which are the frame from which the necessary body parts are to be extracted and the Holistic model to which the desired changes are transferred.

Finally, to determine the position of the controlled body points themselves and their connection to each other for the purpose of clear visualization of the skeleton, an additional function was developed and programmatically implemented to adjust the display subjects (body pose, left hand, right hand) and their characteristics (thickness, size, color).

An example of the implementation of the functions of the third stage for tracking body and hand posture is shown in Fig. 6.

The fourth stage (creation of own dataset for further training of the neural network model). In the course of designing the software implementation, we demonstrated the Sign Language to Text method based on six randomly selected words: ambition, success, but, way, to, sun. For each of the words, 30 videos of 30 frames each were recorded, and for each frame, the position of the pose and hands in the three-dimensional xyz coordinate system was saved.

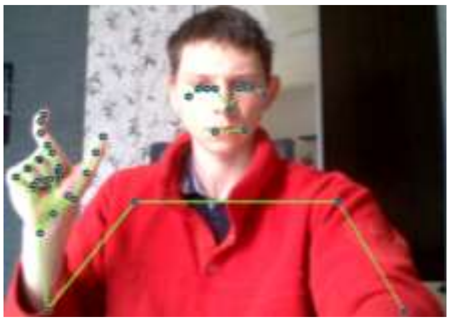

Fig. 6. An example of using the Mediapipe platform to track body and hand posture

The Mediapipe library tracks pose positions at 33 points and hand positions at 21 points. Thus, 258 points were recognized and saved with each frame of the video, and a total of 180 videos were recorded. To complete this stage, the following operations were sequentially implemented:

– creating directories for storing files with the position of body points in space;

– creating a function for extracting the pose, left and right hand position points from the Holistic model and entering them into an integer array;

– recording video of gestures and saving their position in space in files with the npy extension using the point extraction function from step 2 to the corresponding directories generated in step 1.

Thus, as a result of this step, a dataset with points of body position in space was obtained for further training of the neural network. Figure 7 shows an example of recording video of gestures and saving the result to the appropriate directory.

The fifth stage (creating and training a deep learning model). This is a stage of the Sign Language to Text method that requires testing the model for various parameters and determining their optimal values.

To implement this method, it was necessary to choose the types of neural networks that are capable of real-time classification of a sequence of SL image frames. Since each previous and subsequent frame is interconnected, it becomes necessary to use these connections to further train the model and obtain more successful results. Recurrent neural networks (RNNs) are the most suitable for this task, as they contain feedback and allow storing current information and passing it on to the next step of the procedure.

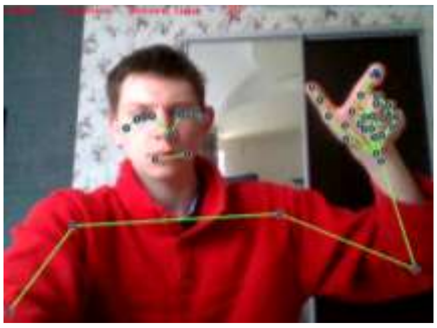

Fig. 7. An example of recording a gesture and saving the position of pose points and hands

The method was implemented and tested using the LSTM architecture (a modification of recurrent neural networks capable of learning long-term dependencies).

The primary dataset was divided into training and test samples in the ratio of 75% to 25%. Next, we built a sequential neural network model consisting of three LSTM layers and three Dense layers. Figure 8 shows the code for building this neural network.

The first LSTM layer has a dimension of 30 by 258 (each video consists of 30 frames for 258 points in the *xyz* coordinate system). The activation function relu was applied to each of the layers except the last one, and the softmax function was applied to the last layer Dense to normalize the results in the range from 0 to 1. During the network training (using the Adam optimizer and the loss function categorical_crossentropy), the number of epochs (110) was determined to minimize the overfitting. Fig. 9 shows a graph of the model's prediction accuracy (horizontal axis) as a function of the number of epochs (vertical axis).

```
model = Sequential()
model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(30, 258)))
model.add(LSTM(128, return_sequences=True, activation='relu'))
model.add(LSTM(64, return_sequences=False, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(actions.shape[0], activation='softmax'))
model.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['categorical_accuracy'])
model.fit(X_train, y_train, epochs=110, callbacks=[tb_callback])
print(model.summary())
```

Fig. 8. Program code for building a sequential model of a recurrent neural network

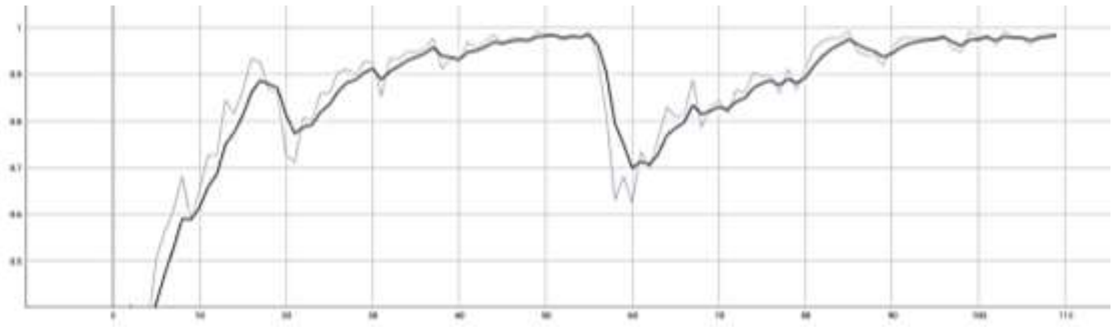

Fig. 9. Graph of forecasting accuracy depending on the number of epochs

The accuracy of the trained and optimized LSTM model is 98.52%. The parameters of this model are shown in Table 2. The results were obtained for the total number of trained parameters in the range (237, 530). Thus, after the fifth stage, a recurrent neural network model that implements the Sign Language to Text method was created, trained, saved, and ready for use.

Table 2. *Parameters of the optimized LSTM model*

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 30, 64)	82688
lstm_1 (LSTM)	(None, 30, 128)	98816
lstm_2 (LSTM)	(None, 64)	49408
dense (Dense)	(None, 64)	4160
dense_1 (Dense)	(None, 32)	2080
dense_1 (Dense)	(None, 6)	198

The sixth stage (visualization of the results). This is the final stage of implementing the Sign Language to text method, which aims to display text classified from gestures in a user-friendly form (ticker) using a trained LSTM model and the OpenCV library.

To improve gesture recognition and correctly display the required text on the screen, a delay of 10 frames after gesture recognition was added. In this way, the user can change the position of their hands and prepare for the next gesture, thereby not interfering with recognition and not adding complexity to the process. An example of the program's operation is shown in Fig. 10.

The Text to Sign Language method is implemented in 4 stages. Below is a description of them.

The first step (tracking body position from video using the Mediapipe library). This stage is similar to the third stage of the Sign Language to Text method, but here the input parameter is a prepared video, not a camera that transmits images in real time. Next, the operations described above are implemented – creating an object to work with the Holistic module, using a function with the desired parameters to track the body, and displaying connections between body points. An example of tracking the position of body points from a video gesture that means "Computer Science" is shown in Fig. 11.

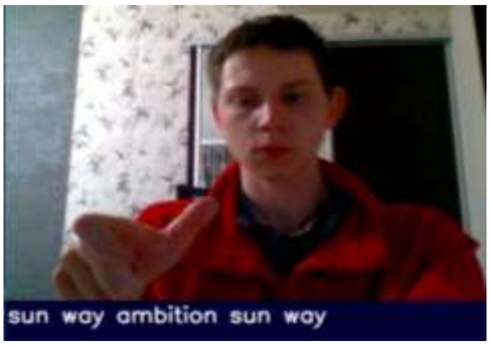

Fig. 10. An example of the final result of the program for the Sign Language to Text method

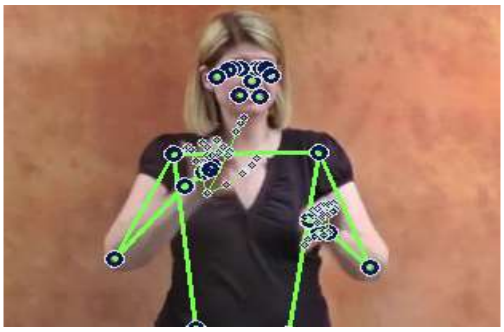

Fig. 11. An example of tracking the position of body points from a video gesture that means "Computer Science"

The second stage (saving pose and hand position points to txt files).

This step is necessary in order to transfer the saved points to the Unity 3D project to generate a 3D character, after which its position will change with each frame in accordance with the transferred points.

It should be noted that when the points are mapped to the Unity 3D project, they may be too close to each other, so their coordinates were multiplied by a thousand and rounded to the third decimal place. This stage allows us to obtain 3 files of body position (right hand, left hand, and pose).

The program code for writing to the pose position file is implemented (writing to the right and left hand position files is done in the same way but using other variables).

The third stage (creating objects in the Unity 3D project).

To create a 3D character, you must first create sphere objects that will be responsible for displaying the corresponding points. To display the pose, only 21 points out of 33 were taken into account, because the gesture is shown above the waist, and therefore the points located at the level of the pelvis, knees, and feet do not carry important information in this case.

To represent the right and left hands, 42 sphere objects were created (21 objects for each hand). To display the connections between the created point-sphere objects, we created line objects connecting one point to another. In this case, 14-line objects connecting the pose points and 21 line objects for the right and left hands were taken into account. Additionally, 6 material colors were added for better visual perception of the graphic: red (body, nose, and mouth pose graphics); blue (ear graphics); green (graphics of connections between points); purple (hand graphics); black (eye graphics); and light brown (background graphics).

The fourth stage (programming of event handlers).

This stage is necessary to create the ability to read files with points in the corresponding classes of event handlers and change the position of sphere and line objects with each frame. Four event handlers were created: BodyAnimationCode, LeftHandAnimationCode, RightHandAnimationCode, LineCode. The first three handlers are designed to read different files (for pose, right and left hands, respectively).

The BodyAnimationCode event handler, using the Start () method, which is called at the beginning of the scene initialization, implements reading a file with points into a list.

In order to change the position of the points with each frame, the Update () method was used. Its idea is to divide the read file line into a comma-separated array, and then to parse each of the array elements, which are the coordinates of the points, into a floating-point type and assign them to the corresponding sphere objects. This sequence of actions is repeated until the file runs out of coordinates of the points being analyzed.

The LineCode () event handler is implemented in a universal form, that is, it is used to display the graphic of the connections of all the necessary points. The idea of this handler is to set the beginning and end of the line in the visual constructor by attaching sphere objects, and using the Update () method to set the beginning and end of the line coordinates. Fig. 12 shows an example of a 3D model created (for displaying a gesture) with an indication of the relationships between points.

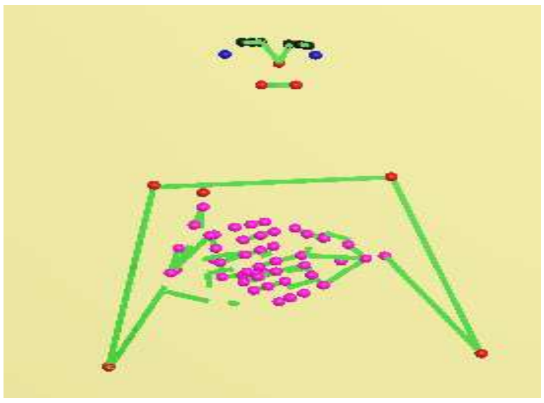

Fig. 12. An example of a 3D model of gesture display with an indication of the connections between points

Conclusions and prospects for further development

The following results were obtained: the main problems and tasks of social inclusion for people with hearing impairments were identified; a comparative analysis of modern methods and technologies of sign language translation was carried out; methods for converting sign language into text (SL-to-Text) and converting text into sign language (Text-to-SL) using neural networks and 3D animation were proposed; the possibility of using the developed methods to automate translation from one sign language to another sign language (Sign Language1 to Sign Language2) was substantiated.

For gesture recognition, a convolutional neural network model is used, which is trained using video gesture datasets imported and generated by the system. The trained model has a high recognition accuracy (98.52%).

The 3D model for displaying the gesture on the screen was created and processed in the Unity 3D environment. The structure of the project, executive and auxiliary files used to build 3D animation for generating sign language concepts includes: event handler files; display results according to which they contain information about the position of the tracked body points; files storing the characteristics of the materials that were added to certain body display points. The Scenes directory was added to the general structure of the automatic sign language translation system, containing system files for setting up the scene, as well as files with body position points for additional gestures used for testing the developed program.

The practical significance of the results of the work is confirmed by the demo versions of the "Sign Language to Text" and "Text to Sign Language" software applications.

The combination of the developed models and technologies makes it possible to implement a comprehensive system of automatic sign language translation that solves a number of the tasks discussed in this paper, namely: converting sign language into text, converting text into sign language and automating translation from one sign language to another sign language using modern intelligent technologies.

A promising area for further research on this topic is the improvement of SL1-to-SL2 methods using different neural network architectures, as well as the creation of open datasets with extended sets of video gestures and concept dictionaries for different sign languages.

References

1. Chumachenko, I.V. "Methods of human resources management in the formation World Health Organization. Deafness and hearing loss". 2023. URL: https://www.who.int/health-topics/hearing-loss (date of application: 10.04.2023).

2. "Web Accessibility Initiative". URL: https://www.w3.org/WAI/ (date of application: 10.04.2023).

3. Decree of the President of Ukraine №722/2019 "On the Sustainable Development Goals of Ukraine for the period up to 2030" dated September 19, 2019. URL: https://www.president.gov.ua/documents/7222019-29825 (date of application: 10.04.2023).

4. Kruglyk, O.P., Horlachev, O.S. (2023), "The importance of auditory perception in the process of sign language translation for people with hearing impairment" [Znachennya slukhovoho spryymannya v protsesi zdiysnennya surdoperekladu dlya osib z porushennyam slukhu], *Scientific journal*. *Series 19 – Correctional pedagogy and special psychology*, No. 43, P. 39–48. DOI; https://doi.org/10.24919/2308-4863/56-2-26

5. Gobhinath, S., Vignesh, T., Pavankumar, R., et al. (2020), "A Study of Hand Gesture Segmentation Techniques for Sign Languages", *Journal of Computational and Theoretical Nanoscience*, Vol. 17, No. 4, P. 1764–1769. DOI: https://doi.org/10.1166/jctn.2020.8439

6. Ghosh, P., Dutta, A., Topno, S. (2022), "Sign Language Hand Glove", *American Journal of Electronics & Communication*, Vol. 3, No. 1, P. 14–16. DOI: https://doi.org/10.15864/ajec.3103

7. Chakraborty, S., Prayosi, P., Sarkar, S., Chakraborty A. (2023), "Sign Language Recognition Using Landmark Detection, GRU and LSTM", *American Journal of Electronics & Communication*, Vol. 3, No. 3, P. 20–26. DOI: https://doi.org/10.15864/ajec.3305

8. Mahesh, R., Kumar, T., Kavin, R, Karthikeyan S. (2020), "Manipulation of Web Using Gestures", *Journal of Computational and Theoretical Nanoscience*, Vol. 17, No.8, P. 3782–3785. DOI: https://doi.org/10.1166/jctn.2020.9320

9. Khetavath, S., Sendhilkumar, N., Mukunthan P., et al. (2023), "An Intelligent Heuristic Manta-Ray Foraging Optimization and Adaptive Extreme Learning Machine for Hand Gesture Image Recognition", *Big Data Mining and Analytics*, No. 6(3), P. 321–335. DOI: https://doi.org/10.26599/BDMA.2022.9020036

10. Zhou, H., Wang, W., Liu G., et al. (2022), "PointGAT: Graph attention networks for 3D object detection", *Intelligent and Converged Networks*, No. 3(2), P. 204–216. DOI: https://doi.org/10.23919/ICN.2022.0014

11. Wuyan, L., Xiaolong, X., Fu, X. (2022), "Human gesture recognition of dynamic skeleton using graph convolutional networks", Journal of Electronic Imaging, Vol. 32, Issue 2, P. 1–21. DOI: https://doi.org/10.1117/1.JEI.32.2.021402

12. John, J., Deshpande, S. (2023), "Hand Gesture Identification Using Deep Learning and Artificial Neural Networks: A Review", *Computational Intelligence for Engineering and Management Applications. Lecture Notes in Electrical Engineering*, Vol. 984, Springer, Singapore. P. 389–403. DOI: https://doi.org/10.1007/978-981-19-8493-830

13. Tan, Y., Lim, K., Lee, C. (2021), "Hand gesture recognition via enhanced densely connected convolutional neural network", *Expert Syst. Appl.*, Vol. 175, No. 114797. P. 28569–28587 DOI: https://doi.org/10.1016/j.eswa.2021.114797

14. Gadekallu, T., Alazab, M., Kaluri, R., et al. (2021), "Hand gesture classification using a novel CNN-crow search algorithm", *Complex & Intell. Syst.*, Vol. 7, P. 1855–1868. DOI: https://doi.org/10.1007/s40747-021-00324-x

15. Zhang, Y., Huang, Y., Sun, X., et al. (2020), "Static and dynamic human arm/hand gesture capturing and recognition via multiinformation fusion of _exible strain sensors", *IEEE Sensors Journal*, Vol. 20, No. 12. P. 6450–6459. DOI: https://doi.org/10.1109/JSEN.2020.2965580

16. Rahim, M., Miah, A., Sayeed A. and Shin J. (2020), "Hand Gesture Recognition Based on Optimal Segmentation in Human-Computer Interaction", Proceedings of the *3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, Kaohsiung, Taiwan, P. 163–166. DOI: 10.1109/ICKII50300.2020.9318870

17. Bhavana, D., Kumar, K., Bipin Chandra, M., et al. (2021), "Hand Sign Recognition using CNN", *International Journal of Performance Analysis in Sport*, Vol. 17(3), P. 314–321. DOI: 10.23940/ijpe.21.03.p7.314321

Received 17.11.2023

Відомості про авторів / About the Authors

Шовковий Євгеній Ігорович – Харківський національний університет радіоелектроніки, студент кафедри штучного інтелекту, Харків, Україна; e-mail: yevhenii.shovkovyi@nure.ua; ORCID ID: https://orcid.org/0009-0007-5613-0946

Гриньова Олена Євгенівна – Харківський національний університет радіоелектроніки, старший викладач кафедри штучного інтелекту, Харків, Україна; e-mail: olena.hrynova@nure.ua; ORCID ID: https://orcid.org/0000-0002-3367-8067

Удовенко Сергій Григорович – доктор технічних наук, професор, Харківський національний економічний університет ім. С. Кузнеця, завідувач кафедри інформатики та обчислювальної техніки, e-mail: serhiy.udovenko@hneu.net; ORCID ID: https://orcid.org/0000-0001-5945-8647

Чала Лариса Ернестівна – кандидат технічних наук, доцент, Харківський національний університет радіоелектроніки, доцент кафедри штучного інтелекту, Харків, Україна; e-mail: larysa.chala@nure.ua; ORCID ID: https://orcid.org/0000-0002-9890-4790

Shovkovyi Yevhenii – Kharkiv National University of Radio Electronics, student at the Department of Artificial Intelligence, Kharkiv, Ukraine.

Grynyova Olena – Kharkiv National University of Radio Electronics, Lecturer at the Department of Artificial Intelligence, Kharkiv, Ukraine.

Udovenko Serhii – Doctor of Sciences (Engineering), Professor, Simon Kuznets Kharkiv National University of Economics, Head at the Department of Informatics and Computer Engineering, Kharkiv, Ukraine.

Chala Larysa – Phd (Engineering Sciences), Associate Professor, Kharkiv National University of Radio Electronics, Associate Professor at the Department of Artificial Intelligence, Kharkiv, Ukraine.

СИСТЕМА АВТОМАТИЧНОГО СУРДОПЕРЕКЛАДУ З ВИКОРИСТАННЯМ НЕЙРОМЕРЕЖНИХ ТЕХНОЛОГІЙ ТА *3D*-АНІМАЦІЇ

Упровадження програмних засобів автоматичного сурдоперекладу в процес соціальної інклюзії людей з вадами слуху є важливим завданням. Соціальна інклюзія для осіб із вадами слуху є нагальною проблемою, яку необхідно вирішувати з огляду на розвиток ІТ-технологій та законодавчі ініціативи, що забезпечують права людей з інвалідністю та їхні рівні можливості. Сказане обґрунтовує актуальність дослідження асистивних технологій у контексті програмних засобів, таких як процес соціального залучення людей з важкими порушеннями слуху в суспільство. **Предметом дослідження** є методи автоматизованого сурдоперекладу із застосуванням інтелектуальних технологій. **Мета роботи** – розроблення та дослідження методів автоматизації сурдоперекладу для поліпшення якості життя людей з вадами слуху відповідно до «Цілей сталого розвитку України» (в частині «Скорочення нерівності»). **Основними завданнями дослідження** є розроблення й тестування методів перетворення жестової мови в текст, перетворення тексту в жестову мову, а також автоматизація перекладу з однієї жестової мови іншою жестовою мовою із застосуванням сучасних інтелектуальних технологій. Для розв’язання цих завдань використовувались **методи** нейромережного моделювання та *3D*-анімації. Унаслідок дослідження здобуто такі **результати**: виявлено основні проблеми й завдання соціальної інклюзії для людей з вадами слуху; здійснено порівняльний аналіз сучасних методів і програмних платформ автоматичного сурдоперекладу; запропоновано й досліджено систему, що об’єднує метод *SL-to-Text*; метод *Text-to-SL* з використанням *3D*-анімації для генерації концептів жестової мови; метод генерації *3D*-анімованого жесту з відеозаписів; метод реалізації технології *Sign Language1 to Sign Language2*. Для розпізнавання жестів застосовано модель згорткової нейронної мережі, що навчається за допомогою імпортованих і згенерованих системою датасетів відеожестів. Навчена модель має високу точність розпізнавання (98,52%). Створення *3D*-моделі для відображення жесту на екран і його оброблення відбувалися у середовищі *Unity 3D*. Структура проєкту, виконавчих і допоміжних файлів, що застосовуються для побудови *3D*-анімації з метою генерації концептів жестової мови, містить: файли обробників подій; результати відображення, що мають інформацію про положення відслідкованих точок тіла; файли, що зберігають характеристики матерій, які були додані до тих чи інших точок відображення тіла. **Висновки:** запропоновані методи автоматизованого перекладу мають практичну значущість, що підтверджують демоверсії програмних застосунків *Sign Language to Text* і *Text to Sign Language*. Перспективним напрямом подальших досліджень з окресленої теми є вдосконалення методів *SL1-to-SL2*, створення відкритих датасетів відеожестів, залучення науковців і розробників для наповнення словників концептами різних жестових мов.

Ключові слова: автоматизація жестового мовлення; анімований персонаж; відслідковування положення тіла; люди з вадами слуху; мова жестів; нейронні мережі; розпізнавання жестів; українська жестова мова; сурдопереклад; скорочення нерівності.

Бібліографічні описи / Bibliographic descriptions

Шовковий Є. І., Гриньова О. Є., Удовенко С. Г., Чала Л. Є. Система автоматичного сурдоперекладу з використанням нейромережних технологій та 3D-анімації. *Сучасний стан наукових досліджень та технологій в промисловості*. 2023. № 4 (26). С. 108–121. DOI: https://doi.org/10.30837/ITSSI.2023.26.108

Shovkovyi, Y., Grynyova, O., Udovenko, S., Chala, L. (2023), "Data structures for deductive simulation of HDL conditional operators", *Innovative Technologies and Scientific Solutions for Industries,* No. 4 (26), P. 108–121. DOI: https://doi.org/10.30837/ITSSI.2023.26.108