DE GRUYTER OPEN
RIGA TECHNICAL UNIVERSITY
Information Technology and Management Science
ISSN 2255-9094 (online)
ISSN 2255-9086 (print)
December 2016, vol. 19, pp. 98–103
doi: 10.1515/itms-2016-0018
https://www.degruyter.com/view/j/itms

Review of Data Preprocessing Methods for Sign Language Recognition Systems based on Artificial Neural Networks

Aleksejs Zorins[1], Pēteris Grabusts[2]
[1, 2] *Rezekne Academy of Technology*

***Abstract* – The article presents an introductory analysis of relevant research topic for Latvian deaf society, which is the development of the Latvian Sign Language Recognition System. More specifically the data preprocessing methods are discussed in the paper and several approaches are shown with a focus on systems based on artificial neural networks, which are one of the most successful solutions for sign language recognition task.**

***Keywords* – Artificial neural networks, sign language recognition.**

I. Introduction

People use gestures as a very important addition to the spoken language; however, each gesture can express much more than a single word. For deaf people the sign language is the main way to communicate with each other and the rest of the world. Sign language gives a possibility for deaf people to be integrated into the society that is why the sign language recognition systems are so important.

The national alphabet is part of sign language system, where each letter is represented by a sign or gesture, usually static, but in many languages, including Latvian, several alphabet letters are shown in motion (see Fig. 1).

The Internet provides several resources, which help people learn the sign language. Among them there is "Spread the Sign" – an international Leonardo da Vinci project within the transfer of innovation programme, which is supported by the European Commission through the Swedish International Programme Office of Education and Training [1]. The project aims to share different sign languages through the Internet.

The website of Latvian Deaf People Rehabilitation has a Sign Language Interpreters' Department, whose main goal is to "facilitate the client's social integration, availability of necessary information and services, provide sign language interpreter's services for communication with other individuals and legal entities according to the client's perception and communication abilities" [2]. The website contains several resources with the Latvian sign language description both in text and video versions.

Despite these examples, Latvia lacks the computerised sign language recognition system, which could help deaf people communicate with each other, people who are unable to see. Such a system could be part of governmental offices, etc. This article continues a previous publication of the authors on this topic [3] and mainly focuses on data preprocessing issues for modelling sign language recognition with artificial neural networks.

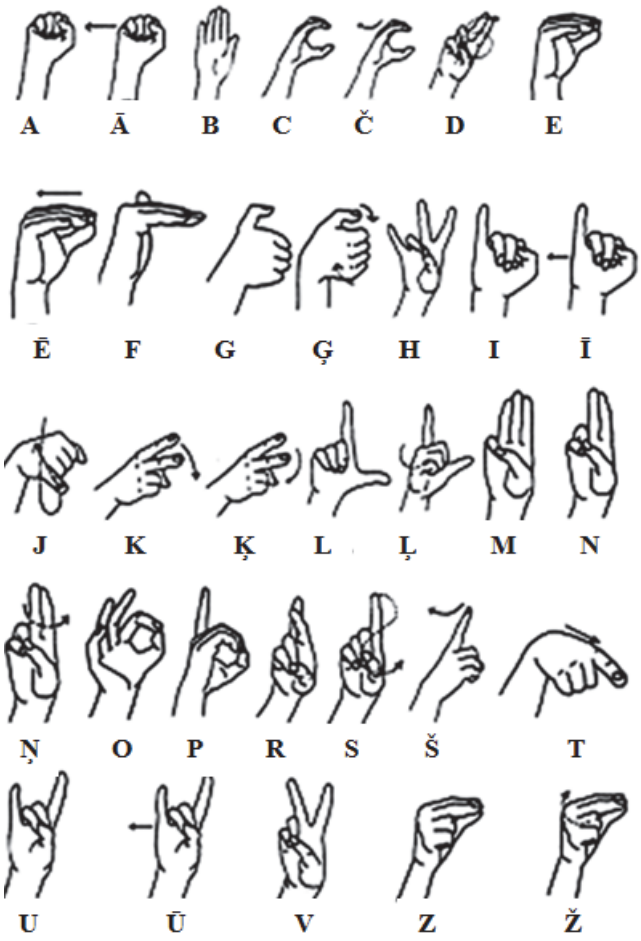


Fig. 1. The symbols of Latvian sign language [2].

The next section presents a short description of sign recognition tasks.

II. Sign Language Recognition Solutions

There are a number of different approaches to the sign recognition task using automated solutions with video and web cameras for correct gesture classification and deaf people language transformation into text or speech.

These approaches are mainly divided into several major groups, depending on input data acquisition technology:

- Use of different markers on hands;
- Implementation of data gloves;
- Infra-red sensor technology (for example, LeapMotion, Microsoft Kinect, etc.);
- Sign recognition via visual (video or photo) methods.

The last two methods are the most popular ones allowing a user to recognise the sign language in real time, capturing gestures through web cameras, Leap Motion or similar technology, whose price is now affordable (Leap Motion sensor costs less than 55 US dollars) [4].

Nowadays sign language recognition is performed by a large number of theoretical and experimental studies using different mathematical and statistical models, as well as applications with hidden Markov models, artificial neural networks, genetic algorithms, etc.

Due to such a variety in classification and recognition algorithms, the choice of an appropriate method for specific sign recognition task takes much time and efforts. The current situation shows several problems in this area related to input data acquisition and preparation as well as to an optimal recognition method.

The next section provides a review of several data preprocessing methods for sign language recognition using artificial neural networks, robust and widely used technology for different task domains [5], [6].

III. Data Preprocessing

Each task, including the sign language recognition, requires proper input data collection and preprocessing. The input data in most applications are collected by web or video cameras [7]. The main reason is affordable costs of hardware.

The data glove technology has several limitations – the less sophisticated gloves give little information about the gesture, at the same time the more advanced technology is not appropriated for a limited budget, for example, Myo Gesture Control Armband costs about 170 British pounds [8]. The data gloves should also be put on and off each time to read the gestures, which poses additional obstacles especially for people with limitations and is not appropriate for installation in public places.

The most perspective technology for real-time sign language recognition at this moment is infra-red sensor technology, such as Leap Motion or Microsoft Kinect. There is a lack of scientific publications in this field that is why this could be a very perspective research direction.

For static pictures such as national alphabets, the video capturing technology is widely used and described in the literature. Let us look at these methods in more detail.

A. Example with Bosnian Language Recognition

We took one of the examples of data acquisition and preprocessing from the paper presenting Bosnian finger spelling alphabet recognition with artificial neural networks [9]. The author used a database of images taken from 3 persons trained to show the finger spelling alphabet. As a result, the database consists of 90 images, 2/3 of which were used for training and 1/3 – for testing the performance of a system.

There are several techniques used to preprocess the data [9]:

- Colour image conversion to grayscale and reduction the size of an image;
- Getting more contrast in an image using histogram equalization;
- Fast Fourier transformation to present an image as an array of frequencies;
- Image binarization to remove background and enhance an object image;
- Detection edge of an image to reduce the amount of data and get only important features;
- Image segmentation using masks for better processing.

Converting an image to a grey scale is a very common procedure, which allows reducing the amount of data to be processed. In this case, one may preserve all important information about gesture. In this example, the authors reduced the size of an image to 375×500 dimensions leaving only the luminance of each pixel.

Histogram equalization enhances the contrast of an image to remove unnecessary artifact that could significantly slow the learning process.

Fast Fourier transform allows presenting an image as a set of frequencies by splitting an image into real and imaginary parts. This enables faster filter application to frequencies rather than to image domain.

The idea of image binarization allows assigning each pixel to class 1 (foreground) or class 0 (background) helping work only with important components and remove the noise of a background.

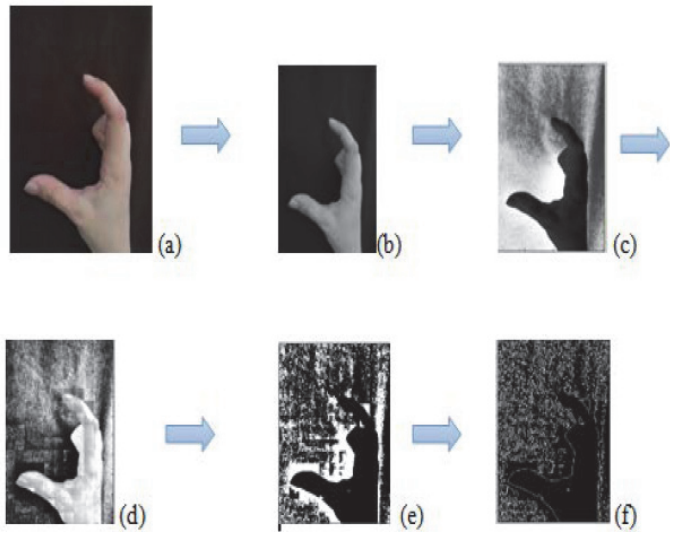


Fig. 2. Image transformation procedure [9].

Image edge detection can be performed using different algorithms, which reduce the amount of data and filter unimportant information from the important one. The authors of this example have chosen the Canny edge detection algorithm. The results of this technique are shown in Fig. 2, which presents image preprocessing starting from an original

image (a); conversion to grey scale (b); histogram equalization (c); Fast Fourier transformation (d); image binarization (e) and, finally, application of Canny edge detection algorithm (f).

B. Example with Real-time Alphabet Recognition

The authors of this example recognised 31 symbols from the American finger spelling alphabet plus 10 digits. All in all, the database consists of 31000 objects from 6 persons participated in the experiments. The 3D camera with the resolution of 240×320 pixels was used to obtain depth images [10].

The authors assumed that the closest object in an image was the hand and then made segmentation of several hand regions using the same depth approach. The algorithm found the connected components from the closest region of the depth image. An additional armband was used for better recognition.

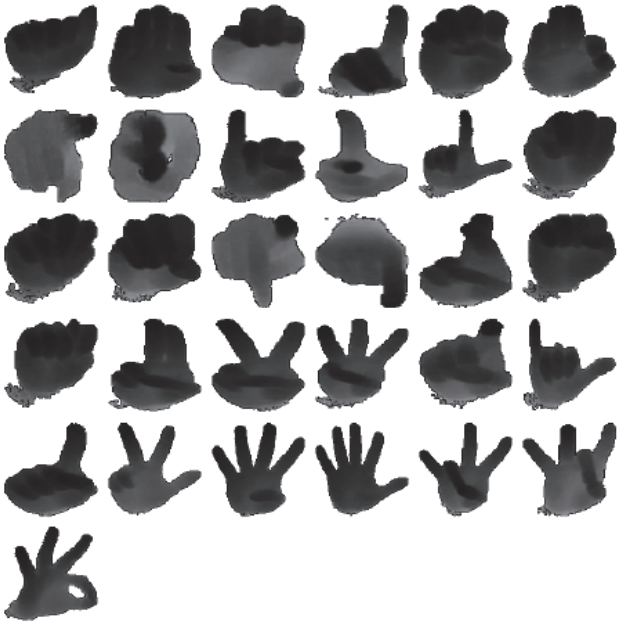


Fig. 3. Examples of preprocessed data consisting of 31 letters and digits [4].

As a result, the bounding box with dimensions of 256×256 was chosen for each palm and included in a final dataset. The recognition phase was performed by the Convolutional Neural Networks in real time by different persons with 85 % average accuracy.

C. Example with Gesture Recognition Using Cosimulation Neural Networks

The next approach also applies the image conversion to a grey scale in the interval from 0 till 255 to recognise the American finger spelling alphabet in real time [11].

Since the images are taken by a camera they are written into text files, which contain the hexadecimal value of the pixels. Each image is stored with its own ID and contains 4096 values.

After that this information is directly read into memory and used for recognition. The performance of such a system has 100 % correctly classified cases for static gestures with an average time of 0.000057 of a second per one image [11].

D. Example of Real-time Sign Language Recognition with Artificial Neural Networks

For this application, the authors tried to recognise the sign language in real time using artificial neural networks and video cameras for image acquisition [12].

The initial preprocessing consists of application of a filtering method based on a moving average or median filter to remove noise from an image. Later the authors of this experiment subtracted the background of an image with the Gaussian average method.

The most interesting phase of preprocessing includes division of the input data into 2 categories: hand shape and hand movement. The hand shape was determined by each fingertip and the direction of movement by the centre of the palm (see Fig. 4).

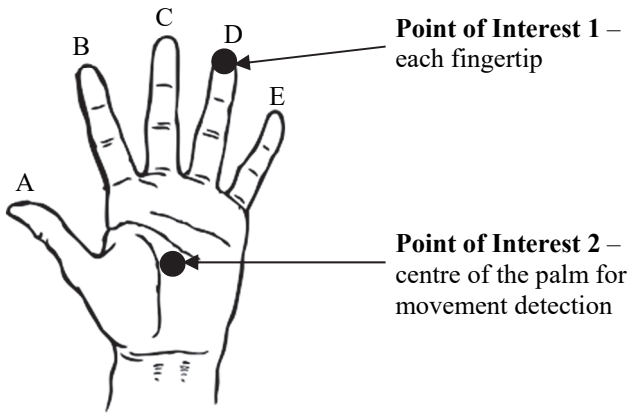
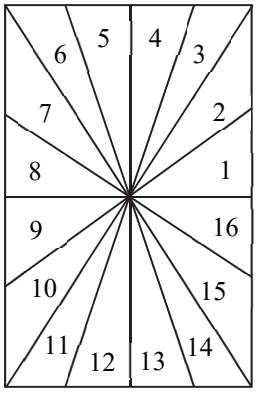


Fig. 4. Points of interest in the palm for better gesture detection adopted by the authors from [12].

These 2 objects in Fig. 4 are called Points of Interest, where each fingertip is used to determine the shape of a gesture and the Point of Interest 2 provides the direction of hand movement in sectors according to Fig. 5.



MV	Direction
1, 16	Horizontal Right/ East
8, 9	Horizontal Left/ West
4, 5	Vertical Up/ North
12, 13	Vertical Down/ South
2, 3	North East
10, 11	South West
6, 7	North West
14, 15	South East

Fig. 5. Motion vector in relation to direction adopted by the authors from [12].

At the end of transformation, the vector of 55 features has been obtained and used to identify each gesture of sign language uniquely. These feature are 5 fingertips, 4 motion vector elements, 6 motion vector sequences and 40 wavelet transformation results of the Fourier transformed image. The Fourier transformation was used to outline the shape of the palm

as the closest curve. These 40 descriptors are Fourier coefficients of object curve.

After preprocessing the recognition with artificial neural networks was applied with 100 % accuracy with noise factor up to 48 %. It means that if even almost half of data is noisy the algorithm still is able to classify an object correctly.

VI. Segmentation Algorithms of Data Series

It is worth mentioning in the context of this article an interesting topic of segmentation algorithms, which allows splitting and limiting data series into a number of segments. This approach is used for time series transformation as well as for other applications [13]. In the case of sign language recognition, one can split the large series of pixels into manageable segments to increase performance of recognition algorithm.

Some of the transformation algorithms have been tested by the author in [14]. These are sliding window, top-down and bottom-up algorithms. The next figures show algorithm flowcharts.

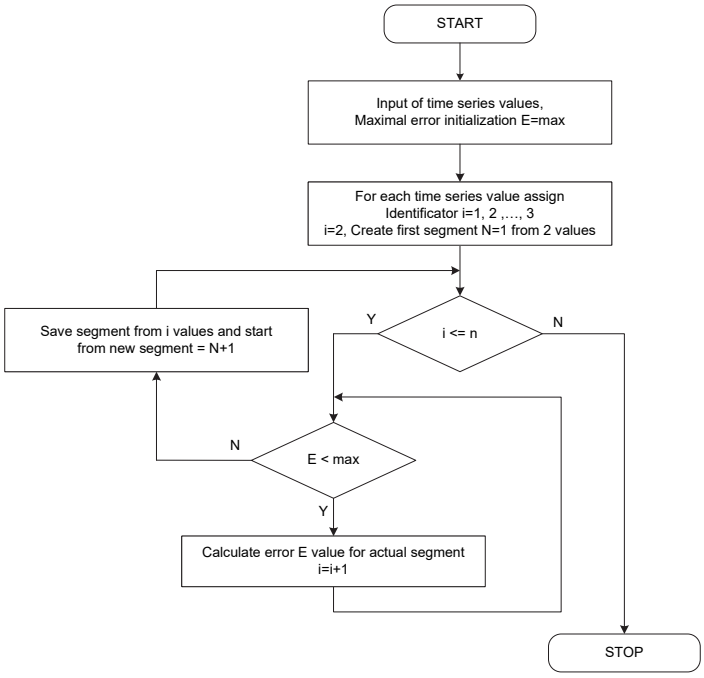


Fig. 6. Sliding window algorithm flowchart (adopted by the authors from [13].

Sliding window algorithm takes the first series data point, increasing size of a segment and calculating approximation error for each possible kind of a segment. When the maximum error threshold is reached, the algorithm makes segment from i-1 series data points and the process repeats until all values are transformed into segments.

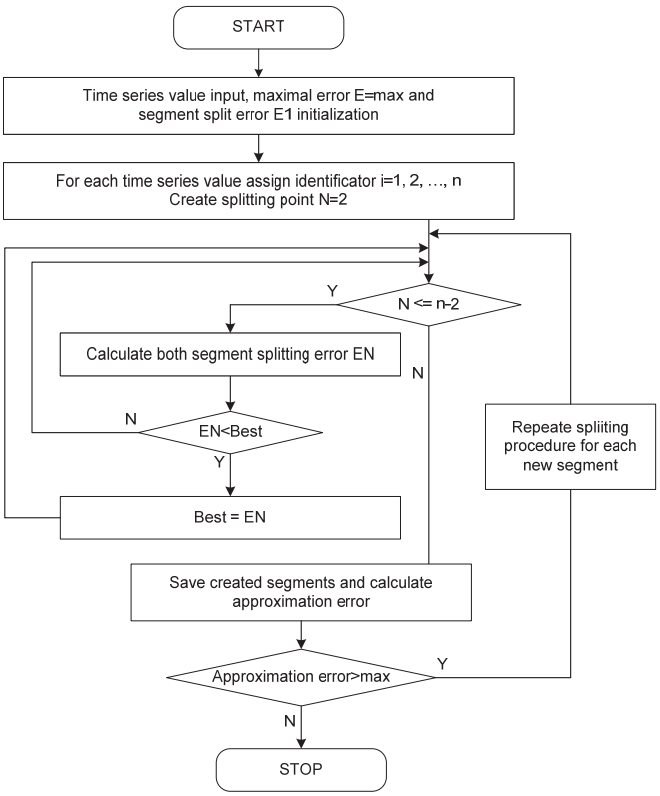


Fig. 7. Top-down algorithm flowchart (adopted by the authors from [13].

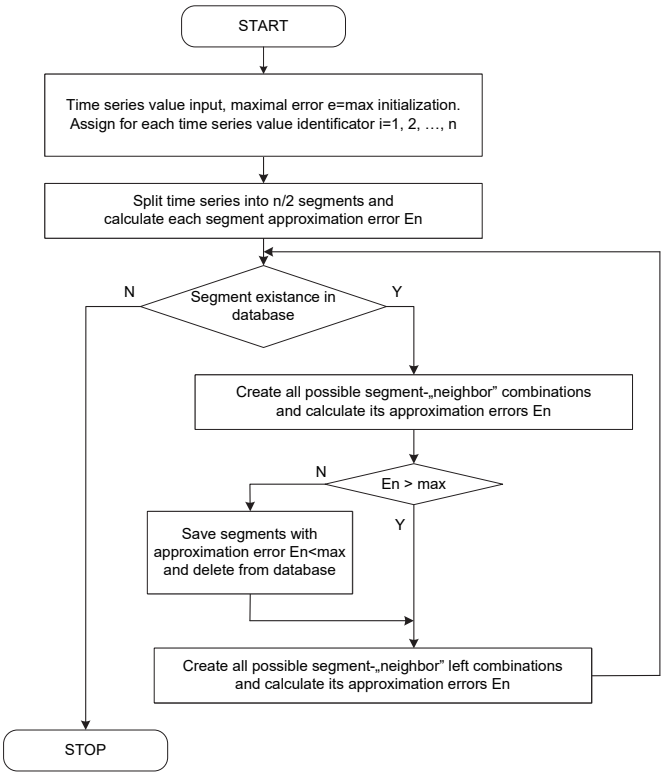


Fig. 8. “Bottom-up” algorithm flowchart (adopted by the authors from [13].

The sliding window algorithm is rather simple and allows its implementation in real time.

The "Top-Down" segmentation algorithm (Fig. 7) initially splits the data series into 2 big equal segments, and then analyses each of them, calculating an approximation error. Each segment can be split as many times as necessary to reach the desired threshold.

The "Bottom-Up" algorithm is shown in Fig. 8. It goes in the opposite way initially splitting all data into small segments and then combining these segments according to the error threshold.

All three algorithms help perform piece-wise linear approximation, which could be calculated as follows:

- linear interpolation (method simply connects two endpoints of a segment);
- linear regression (segment approximation line is obtained using least square regression).

Linear interpolation takes less calculation and consequently less processing time; therefore, it is better for on-line usage, when time consideration is foremost. Linear regression is more time consuming, but allows obtaining better precision.

Linear interpolation simply connects 2 end points of a segment, at the same time linear regression is calculated as follows:

$$X_{i+1} = a + bX_i, \tag{1}$$

where a and b are approximation line coefficients. Value b is calculated as follows:

$$b = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}, \tag{2}$$

where

y_i – independent variable;

x_i – dependent variable;

n – number of data points.

After obtaining information about segments, one can refer to an approach presented in [14].

TABLE I

DESCRIPTION OF NEURAL NETWORK INPUTS IN THE CASE OF DATA SERIES SEGMENTATION

Identifier label	Description
AC	Linear regression coefficient *a* for each approximated segment
BC	Linear regression coefficient *b* for each approximated segment
LS	Length of a segment (in data points)
SS	Starting point of a segment
ES	End point of a segment
CL	Class identifier

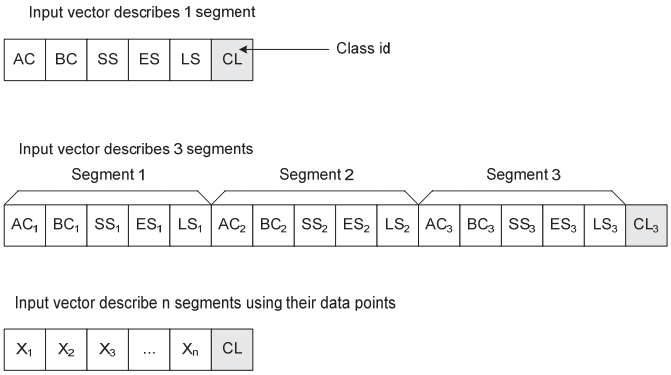


Fig. 9. Some possible input vectors for neural network.

The data series transformation results can be treated to a neural network or another recognition algorithm in two ways:

- Using real data series values from created segments. This approach is similar to a windowing approach, but the length of segments can be different.
- Using as input data from Table I.

As a result, the input vectors used in training may be different. Some of possible solutions are given in Fig. 9 [14].

V. APPLICATIONS WITH ARTIFICIAL NEURAL NETWORKS

After data preparation described above, the next step of sign language recognition is actually recognition itself when a neural network is trained on the test samples (training set) of signs and then the system is used for unknown (not presented in the training phase) data recognition.

In Example A from the previous section, a back-propagation training algorithm with sigmoid activation function and two hidden layers was used. This neural network had 15 input neurons and 90 training samples. As a result, the authors of the article obtained 85 % correctly classified patterns with 40 neurons in a hidden layer [9].

Kang in Example B used Convolutional Neural Networks, particularly Caffe implementation (CaffeNet). The network consisted of 5 convolution layers, 3 max-pooling layers, and 3 fully connected layers [10].

Mekala and his team (Examples D) used a combinational artificial neural network, which was based on the cache search memory concept of a CPU when all blocks of a system had a dual information exchange bus [12]. The recognition part was also based on error back-propagation. Another work of this author (Example C) used cosimulation neural networks, when part of the neural network was implemented on the hardware with dedicated ports and different levels of the neural network were communicating with one another on two different platforms [11]. Such a network had 16 input neurons, 50 neurons in the first hidden layer and 50 in other hidden layer, and 35 output neurons.

VI. Conclusion and Future Research

The data preprocessing methods presented here give a solid background for further experiments. Most solutions have no problems with static and dynamic gesture alphabet recognition while more advanced ones recognise additional deaf language signs and words in real time.

Despite a good number of successful applications in sign language recognition with Artificial Neural Networks there is still a lack of automated solution for the Latvian sign language. Sign language recognition system for the Latvian language requires several major steps:

- Creation of a database of the Latvian sign language;
- Selection and implementation of appropriate data pre-processing method;
- Selection of proper recognition algorithm and training the neural network to obtain confident results (example D along with the segmentation method presented in Section III are the most perspective for further experiments);
- Development of software for the Latvian deaf community.

These are the main directions of future research by the authors to be conducted in the near future with the aim to help people with special needs to integrate into our society.

References

[1] Spread the sign. [Online]. Available: https://www.spreadthesign.com/us/aboutus/ [Accessed: May 11, 2016].

[2] The Latvian Sign Language Development Department. [Online]. Available: http://rc.lns.lv/index.php [Accessed: May 11, 2016].

[3] A. Zorins and P. Grabusts, "Review of sign language recognition systems based on artificial neural networks," in *MENDEL 2016 conference proceedings,* Brno, Czech Republic, 2016.

[4] The Leap Motion Store. [Online]. Available: http://store-us.leapmotion.com/ [Accessed: May 11, 2016].

[5] L. Fausett, *Fundamentals of Neural Networks. Architectures, algorithms and applications*, Upper Saddle River, NJ: Prentice-Hall, 1994.

[6] R. Rojas, *Neural networks. A systematic approach*, Berlin, Germany: Springer, 1996.

[7] H. Cooper, B. Holt and R. Bowden, "Sign Language Recognition," in *Visual Analysis of Humans: Looking at People*, London, UK: Springer, 2011, pp. 539–562. https://doi.org/10.1007/978-0-85729-997-0_27

[8] Myo Gesture Control Armband. [Online]. Available: https://www.amazon.co.uk/MYO-MYO-00002-001-Gesture-Control-Armband/dp/B00O66E58M/ref=sr_1_1?ie=UTF8&qid=1462967348&sr=8-1&keywords=myo [Accessed: May 11, 2016]

[9] S. Đogić and G. Karli, "Sign Language Recognition using Neural Networks," *TEM Journal*, vol. 3, issue 4, pp. 296–301, 2014.

[10] B. Kang, S. Tripathi and T. Nguyen, "Real-time Sign Language Fingerspelling Recognition using Convolutional Neural Networks from Depth map," in *3rd IAPR Asian Conference on Pattern Recognition,* Kuala Lumpur, Malaysia, 2015. https://doi.org/10.1109/acpr.2015.7486481

[11] P. Mekala et al., "Gesture Recognition Using Neural Networks Based on HW/SW Cosimulation Platform," *Advances in Software Engineering*, vol. 2013, 2013. https://doi.org/10.1155/2013/707248

[12] P. Mekala et al., "Real-time Sign Language Recognition based on Neural Network Architecture," in *IEEE 43rd Southeastern Symposium on System Theory*, Auburn, AL, 2011, pp. 195–199. https://doi.org/10.1109/SSST.2011.5753805

[13] E. Keogh and M. Pazzani, "An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback," in *Proceedings of the Fourth International Conference of Knowledge Discovery and Data Mining*, 1998, pp. 239–241.

[14] A. Zorins, "Data preprocessing methods for interval based neural network prediction," in *Proceedings of 7th International Scientific Practical Conference "Environment. Technology. Resources,"* Rezekne, Latvia, 2007.

Aleksejs Zorins received his Mg. sc. ing. degree in Information Technology from Riga Technical University in 2001. Since that time he has working at Rezekne Academy of Technologies as a Lecturer. From 2009 till 2016 he was the Director of the study programme "Electronic Commerce" at the Department of Computer Science. Since 2015 he has been a Doctoral student at Rezekne Academy of Technologies, Faculty of Engineering.
His research interests include artificial neural networks, data mining technologies and clustering methods. His current research interests include applications of artificial neural networks to sign language recognition.
E-mail: Aleksejs.Zorins@ru.lv

Pēteris Grabusts received his Dr. sc. ing. degree in Information Technology from Riga Technical University in 2006. Since 1996 he has been working at Rezekne Academy of Technologies. Since 2014 has been a Professor at the Department of Computer Science.
His research interests include data mining technologies, neural networks and clustering methods. His current research interests include ontologies and techniques for clustering.
E-mail: peter@ru.lv