DOI: 10.5604/01.3001.0013.2542

CROSS PLATFORM TOOLS FOR MODELING AND RECOGNITION OF THE FINGERSPELLING ALPHABET OF GESTURE LANGUAGE

Serhii S. Kondratiuk[1], **Iurii V. Krak**[1,2], **Waldemar Wójcik**[3]
[1]Taras Shevchenko National University of Kyiv, [2]Glushkov Institute of Cybernetics of NAS of Ukraine, [3]Lublin University of Technology

Abstract. *A solution for the problems of the finger spelling alphabet of gesture language modelling and recognition based on cross-platform technologies is proposed. Modelling and recognition performance can be flexible and adjusted, based on the hardware it operates or based on the availability of an internet connection. The proposed approach tunes the complexity of the 3D hand model based on the CPU type, amount of available memory and internet connection speed. Sign recognition is also performed using cross-platform technologies and the tradeoff in model size and performance can be adjusted. the methods of convolutional neural networks are used as tools for gestures of alphabet recognition. For the gesture recognition experiment, a dataset of 50,000 images was collected, with 50 different hands recorded, with almost 1,000 images per each person. The experimental researches demonstrated the effectiveness of proposed approaches.*

Keywords: cross platform, sign language, fingerspelling alphabet, 3D modeling, Convolutional Neural Networks

CROSS-PLATFORMOWE NARZĘDZIA DO MODELOWANIA I ROZPOZNAWANIA ALFABETU PALCOWEGO JĘZYKA GESTÓW

Streszczenie. *Zaproponowano rozwiązanie problemów z alfabetem daktylograficznym w modelowaniu języka gestów i rozpoznawaniu znaków w oparciu o technologie wieloplatformowe. Wydajność modelowania i rozpoznawania może być elastyczna i dostosowana, w zależności od wykorzystywanego sprzętu lub dostępności łącza internetowego. Proponowane podejście dostosowuje złożoność modelu 3D dłoni w zależności od typu procesora, ilości dostępnej pamięci i szybkości połączenia internetowego. Rozpoznawanie znaków odbywa się również z wykorzystaniem technologii międzyplatformowych, a kompromis w zakresie wielkości modelu i wydajności może być dostosowany. Jako narzędzia do rozpoznawania gestów alfabetu wykorzystywane są metody konwolucyjnych sieci neuronowych. Na potrzeby eksperymentu rozpoznawania gestów zebrano zbiór danych obejmujący 50 000 obrazów, przy czym zarejestrowano 50 różnych rąk, a na każdą osobę przypadało prawie 1000 obrazów. Badania eksperymentalne wykazały skuteczność proponowanego podejścia.*

Słowa kluczowe: cross platform, język migowy, alfabet palcowy, modelowanie 3D, konwolucyjne sieci neuronowe

Introduction

Gesture based communication is one of real methods for data transition, close by with content and discourse. Signs can be utilized to define explicit letters, words, states and can be handled, encoded and put away in a different ways. Building up a technology for storing, modeling and demonstrating signs and communications via gestures is a challenging issue because of contrasts in accessible platforms. Different platforms have different working operating systems, (for example, mobile – iOS, Android, desktop – MacOS, Linux, Windows, and web – ChromeOS, and so forth), which infers diverse execution level and requires porting the codebase on every stage; some platforms require web connection, (for example, distributed computing technologies [10]) and others don't, and so forth. Displaying such a technology for sign language is a real issue for individuals with hearing disabilities and their relatives, yet in addition is significant in a more extensive usage, due to universality of sign language.

Cross-platform development [17] give an approach to beat this issue. Cross-platform development can be utilized instead of virtual-machines [15] or a lot of mono-platforms development. Utilizing these advances permits to build up a single codebase for various sort of platforms, types of CPU, operating systems of equipment execution and to send it on all platforms consistently.

In this article an answer for the issue of sign language demonstrating is proposed dependent on cross-platform development. The technology of communication through signs can be adaptable and balanced, depending on the equipment it works on or dependent on accessibility of internet connection. The proposed methodology tunes the 3D hand model (parameters, for example, the quantity of polygons for rendering the hand and the step of signs progress) in view of the CPU type, measure of accessible memory and web connection speed. The sign recognition is additionally performed utilizing cross-platform developments and can be alter ed for the tradeoff in model size and execution speed. The sign (gesture) modeling and recognition is a part of a single gesture communication technology and this paper is a further development of author's previous works [5, 7].

1. Existing approaches for modeling and recognition of sign language and their implementation on different platforms

A technology for both sign language displaying and recognition can be considered as a pair of modules for gesture demonstrating and gesture recognition. Some of frameworks give just a single module, regularly just for a particular platform. American Sign Language Online Dictionary [2] is one of the systems which were collected for signal displaying, it depends on a lot of recorded recordings, stored in a database, and this methodology was used in a lot of organizations [1]. In any case, because of its strategy for storing motions and no conceivable capacity to adjust them, this framework isn't adaptable and constrained distinctly to a lot of pre-recorded records. Likewise, no gesture recognition method is pro-vided.

Three-dimensional hand model is a significant piece of gesture language displaying. Two gatherings of hand demonstrating approaches are investigated in the work [4]. In the paper, spatial approach considers situating of the hand sign and their parametrical set. The methodology depends on the rules of advances of a sign.

In the paper [11] builds up an approach for demonstrating motion for an input content. The technology comprise of a factual model for given input handling and a generative calculations for ap-propriate hand motion displaying, utilizing indicated kinematics. As a result of the work, authors provide ANVIL tools for an-notation, DANCE library for sign transition and sign generator NOVA [14]. However, the technology is specified to work only on Windows operating system and x86 CPU. Comparative technology for displaying signs is proposed in [9], but additionally just for a single platform.

2. Problem statement

The proposed technology should comprise of two sections, which are communication through signs [8, 9] displaying and recognition module. The two modules ought to have the option to keep running without codebase alteration on various platforms and ought to be created utilizing cross-platform developments.

Sign recognition module should comprise of a model which can recognize and distinguish the gesture, determined by the client, from a camera input. Set of gestures is constrained by the Ukrainian dactyl language, however can be broadened further. A fitting dataset of Ukrainian dactyl language ought to be gathered for testing the model execution. The sign language displaying module ought to have the option to recreate a gesture specified by a lot of parameters, stored in a database, and ought to be restricted by a lot of Ukrainian dactyl language signs, yet can be expanded further with different languages. The gesture displaying module ought to likewise have the option to show gesture motions, which means it can demonstrate flawlessly words and sentences, comprising of Ukrainian dactyl language signs.

3. Proposed approach

To built up a technology for fingerspelling letters in order to demonstrate and recognize gestures, which can keep running on numerous platforms, without changing the codebase, a methodology based on cross-platform instruments is proposed. Gesture displaying module should comprise of a virtual three dimensional hand model and a user interface (UI), which ought to furnish the client with ability to input a sequence of letters, which at that point will be changed into a sequence of gestures. To implement both hand model and UI, a cross-platform framework Unity3D [18] was utilized. Contrasting with other 3D engine, it provides a unified development process for all available platforms (mo-bile, desktop and web) and provides a seamless way to deploy the application on all of them without changing the codebase. To build up a sign recognition module, a cross-platform tool Tensorflow [16] is proposed. This methodology based on cross-platform tools for AI permits to created and train a sign recognition model once, and after that convey it on different platforms (portable, desktop and web) with no changes to the model or the code for training. As a model engineering, the MobileNet design is considered, upgraded with 3D convolutions, to take into account data from a grouping of frames from the camera. Overall, the proposed approach novelty is that it assembles together cross-platforms technology for Ukrainian dactyl language displaying and recognition, with improved MobileNet design for improved recognition of the Ukrainian dactyl letters.

4. Infologic model

The framework design chart (Fig. 1) shows the communication of fundamental parts of the proposed technology.

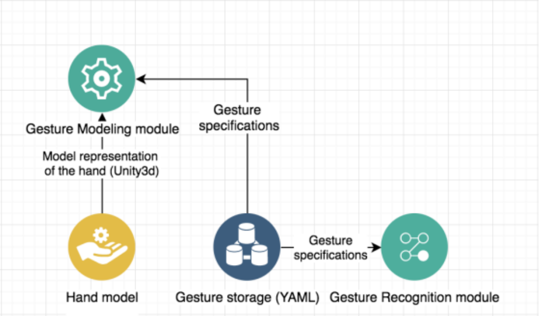


Fig. 1. Infologic model of cross platform gesture communication technology

The signs for gesture demonstrating are stored in a predefined format (YAML [16]) in a database, and are used by the gesture displaying engine for setting an arrangement of a spatial three-dimensional hand model utilizing indicated parameters for the motion from a database section. The sign modeling module works over the gesture database and is a piece of the application, which comprises of gesture displaying and UI parts, them two being created with Unity3D system, utilizing C# programming language. The virtual hand model is determined by a skeleton and a lot of parameters and their restrictions for every skeleton joint. The sign recognition module is executed with Tensorflow system, using Python programming language. The sign recognition module run autonomously of gesture demonstrating module and database. Primary segments of the gesture recognition module is the model which performs gesture recognition and the wrapper which changes over camera input to appropriate data for the model.

5. Gesture modeling

The three-dimensional hand model skeleton was implemented in view of human hand structures. Skeleton model comprises of: 8 bones in wrist, 3 bones in the thumb and 1 metacarpus and 3 phalanges in every one of different fingers. Each joint of each pair of bones has it's own sort of association and it's own parameters for setting this joint, it's very own level of freedom and it's restrictions. Generally speaking, the hand model is represented with a skeleton which comprises of 27 bones and has 25 degrees of portability. The thumb has 5 degrees of freedom, middle and pointers have four degrees of freedom, four degrees of freedom are situated in the metacarpal-carpal joint to the little finger and thumb to empower development of the palm.

Unity3D framework was utilized for developing the three dimensional hand model, since building up your very own cross-platform rendering engine is a non-trivial assignment. Unity3D was chosen because of friendly UI, capacity to actualize through it's methods both the scene and UI. Over the hand skeleton, a sensible hand model was created, rendered with in excess of 70,000 polygons (Fig. 2), Unity3D system can deal with such model with satisfying execution.

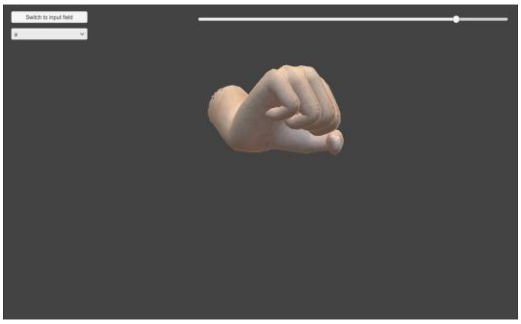


Fig. 2. Gesture modeling under iOS platform

6. Gesture recognition

Gesture learning and gesture recognition modules, developed with cross platform tools (frameworks based on Python, C++) can be embedded into information and gesture communication cross platform technology. Multiple approaches were considered as an approach for gesture recognition. Automatic sign language recognition can be approached similarly to speech recognition, with signs being processed similar to phones or words. Conventionally, sign language recognition consists of taking an input of video sequences, extracting motion features that reflect sign language linguistic terms, and then using pattern mining techniques or machine learning approaches on the training data.

Convolutional Neural Networks (CNNs) [12] have shown robust results in image classification and recognition problems, and have been successfully implemented for gesture recognition in recent years. In particular, deep CNNs have been used in researches done in the field of sign language recognition, with input-recognition that utilizes not only pixels of the images. With the use of depth sense cameras, the process is made much easier via developing characteristic depth and motion profiles for each sign language gesture. Multiple existing researches done over various sign languages show that CNNs achieve state-of-the-art accuracy for gesture recognition [6, 12].

Convolutional neural networks have such advantages: no need in hand crafted features of gestures on images; predictive model is able to generalize on users and surrounding not occurring during training; robustness to different scales, lightning conditions and occlusions. Although, selected approach has couple of disadvantages, which may be overcome with a relatively big dataset (1,000 images for each gesture, among more than 10 people of different age, sex, nationality and images taken under different environment conditions and scales): need to collect a rather big and labeled gesture images dataset; black-box approach which is harder to interpret. Usage of cross platform neural network framework such as Tensorflow allows to implement gesture recognition as a cross platform module of proposed technology and serve trained recognition model on server or transfer it to the device [3].

For experiment there was collected (Fig. 3) a dataset with Ukrainian dactyl language letters. Each gesture consists of 1500 sample images, and 50 different people hands were showing gestures, with distribution of 70% male and 30% female hands. Different light conditions were used (with distribution of 20% images in bad light conditions, 30% in mediocre light conditions and 50% in good light conditions). About 10% of images were distorted with noise and blur.

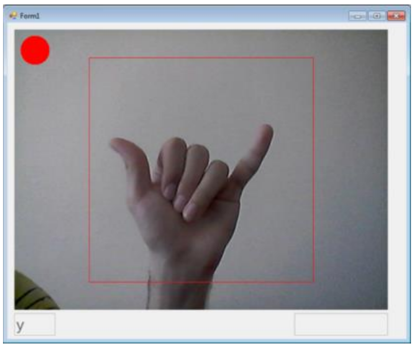


Fig. 3. UI tool for collection gesture database

MobileNet [6] architecture was used as a basis for CNN architecture. It has multiple advantages, such as good trade-off on accuracy and performance, especially on mobile devices, which are aimed to use, as the technology is cross-platform. The MobileNet model is based on depth wise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depth wise convolution and a 1×1 convolution called a point wise convolution. For MobileNets the depth wise convolution applies a single filter to each input channel. The point wise convolution then applies a 1×1 convolution to combine the outputs the depth wise convolution. A standard convolution both filters and combines inputs into a new set of outputs in one step. The depth wise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size.

Process of training MobileNet network for gesture detection takes ~ 200.000 iterations, which is approximately 10 epochs. Figure 4 shows example UI of how the proposed technology detects specific gesture from Ukrainian dactyl and draws a bounding box over a detected gesture.

7. Gesture recognition experiment

For the training process of MobileNet architecture based Convolutional Neural Network for the task of gesture recognition of Ukrainian dactyl alphabet gestures an appropriate dataset should have been collected, due to no available datasets for Ukrainian sign language in free access. A specific software was developed for recording a short video sequences of Ukrainian dactyl alphabet gestures shown by different people. Since the recording software isn't direct part of the proposed technology, but rather a helper tool, it was developed only under Windows family of operating systems, using C# programing language and .NET framework. The pipeline of recording a single entry looks like this:

The person sits in front of the webcam, connected to the recording software;

The person needs to put one's hand into the region of interest of the recording software;

The person shows specific gesture from the Ukrainian dactyl alphabet;

The recording operator starts the recording;

The person showing the gesture starts to smoothly move the hand across different axis's;

After video of appropriate length was recorded, the operator stops the recording;

The process goes on with the next gesture.

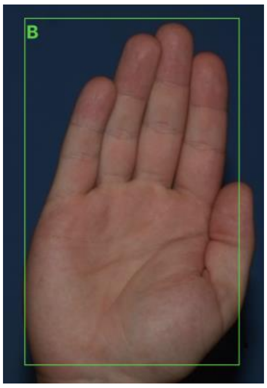
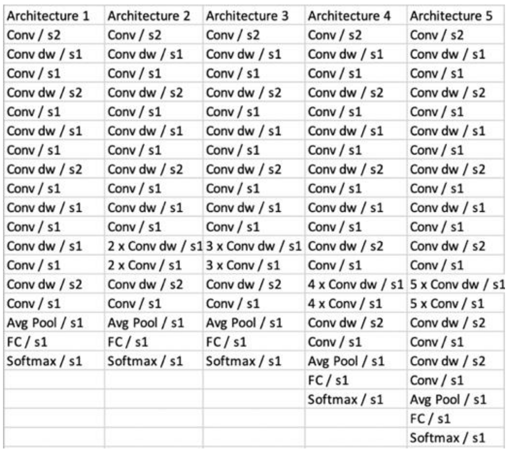


Fig. 4. Example of recognition UI

8. Dataset and model architecture

Since training of the Convolution Neural Network hardly depends on a big and diverse dataset, to achieve a high enough accuracy metrics level, dataset of Ukrainian dactyl language letters with diverse characteristics was collected. More than 50,000 original images were collected as a training dataset. After applying additional dataset augmentation techniques (such as rotation, random crop, mirroring etc.) the final dataset became about 150,000 images. For testing purposes a fraction of 10% of the dataset was selected, making final training dataset of 135,000 images and final testing dataset of 15,000 images.

Table 1. Different architectures trained



Architecture 1	Architecture 2	Architecture 3	Architecture 4	Architecture 5
Conv / s2	Conv / s2	Conv / s2	Conv / s2	Conv / s2
Conv dw / s1	Conv dw / s1	Conv dw / s1	Conv dw / s1	Conv dw / s1
Conv / s1	Conv / s1	Conv / s1	Conv / s1	Conv / s1
Conv dw / s2	Conv dw / s2	Conv dw / s2	Conv dw / s2	Conv dw / s2
Conv / s1	Conv / s1	Conv / s1	Conv / s1	Conv / s1
Conv dw / s1	Conv dw / s1	Conv dw / s1	Conv dw / s1	Conv dw / s1
Conv / s1	Conv / s1	Conv / s1	Conv / s1	Conv / s1
Conv dw / s2	Conv dw / s2	Conv dw / s2	Conv dw / s2	Conv dw / s2
Conv / s1	Conv / s1	Conv / s1	Conv / s1	Conv / s1
Conv dw / s1	Conv dw / s1	Conv dw / s1	Conv dw / s1	Conv dw / s1
Conv / s1	Conv / s1	Conv / s1	Conv / s1	Conv / s1
Conv dw / s1	2 x Conv dw / s1	3 x Conv dw / s1	Conv dw / s2	Conv dw / s2
Conv / s1	2 x Conv / s1	3 x Conv / s1	Conv / s1	Conv / s1
Conv dw / s2	Conv dw / s2	Conv dw / s2	4 x Conv dw / s1	5 x Conv dw / s1
Conv / s1	Conv / s1	Conv / s1	4 x Conv / s1	5 x Conv / s1
Avg Pool / s1	Avg Pool / s1	Avg Pool / s1	Conv dw / s2	Conv dw / s2
FC / s1	FC / s1	FC / s1	Conv / s1	Conv / s1
Softmax / s1	Softmax / s1	Softmax / s1	Avg Pool / s1	Conv dw / s2
			FC / s1	Conv / s1
			Softmax / s1	Avg Pool / s1
				FC / s1
				Softmax / s1

Standard techniques of fighting overfitting of the neural network were applied on each training. Different architectures (Table 1) and their metrics and confusion matrixes are shown. Architecture 5 stopped showing growth in f1 score although having more complex performance. Architecture 4 (Fig. 5) was selected as the final option for the proposed technology as the best tradeoff of architecture size to performance.

During the training process of MobileNet architecture based Convolutional Neural Network multiple architecture modifications were set up in order to find the best trade-off in number of layers to accuracy. At some point the accuracy of the trained model stopped increasing, which is show in Fig. 5 so the architecture No 4 as decided as optimal in terms of the smallest architecture with best accuracy (macro average f1-score).

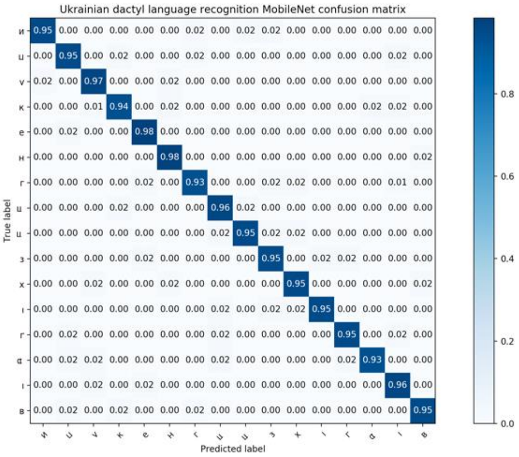


Fig. 5. Confusion matrix of the best trained architecture No 4

9. Conclusions

The proposed technology consists of two main modules: gesture modeling and gesture recognition modules, which use the database with gestures specifications stored in YAML format in a PostgreSQL database.

The proposed technology implements gesture modeling and gesture recognition for Ukrainian dactyl alphabet gestures with cross-platform development tools. Gesture modeling was implemented using Unity3D framework, which is cross-platform and shows satisfying performance on different platforms (mobile, web and desktop) while rendering a realistic three-dimensional hand model. Number of polygons and animation step of gesture transitions can be adjusted for the sake of performance.

A dataset of more than 50.000 images was collected using diverse conditions and different persons hands. The dataset was augmented using specific techniques and final dataset consists of 150.000 images. Gesture recognition module was implemented using Tensorflow framework, which provides ability to deploy its model on different platforms without any codebase modifications. As a model for gesture recognition, MobileNet architecture was chosen, as a model with best trade-off of size and accuracy, especially on low performance platforms (such as mobile and web). The model was trained on the collected Ukrainian dactyl language dataset. Due to augmentations, the model showed state-of-the-art level of performance. Based on experiments, optimal model architecture was chosen in order to keep the best performance level with the least model size possible. According experiments results were shown. The performance of CNN model was compared to other approaches and showed similar or superior values.

The proposed gesture communication technology can be further augmented with other gestures and languages and with other cross-platform modules.

References

[1] Apple Touchless Gesture System for iDevices http://www.patentlyapple.com/patently-apple/2014/12/apple-invents-a-highly-advanced-air-gesturing-system-for-future-idevices-and-beyond.html (available 15.05.2019).

[2] ASL Sign language dictionary http://www.signasl.org/sign/model (available 15.05.2019).

[3] Howard A.G., Wang W.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications https://arxiv.org/pdf/1704.04861.pdf (available 15.05.2019).

[4] Khan R.Z., Ibraheem N.A., Meghanathan N., et al.: Comparative study of hand gesture recognition system. SIPM, FCST, ITCA, WSE, ACSIT, CS & IT 06/2012, 203–213.

[5] Krak I., Kondratiuk S.: Cross-platform software for the development of sign communication system: Dactyl language modelling, Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 1/2017, 167–170 [DOI: 10.1109/STC-CSIT.2017.8098760].

[6] Krizhevsky I. Sutskever, Hinton G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 2012, 1097–1105.

[7] Kryvonos I.G., Krak I.V., Barchukova Y., Trotsenko B.A.: Human hand motion parametrization for dactylemes modeling. Journal of Automation and Information Sciences 43(12)/2011, 1–11.

[8] Kryvonos I.G., Krak I.V., Barmak O.V., Shkilniuk D.V.: Construction and identification of elements of sign communication. Cybernetics and Systems Analysis 49(2)/2013, 163–172.

[9] Kryvonos I.G., Krak I.V.: Modeling human hand movements, facial expressions, and articulation to synthesize and visualize gesture information. Cybernetics and Systems Analysis 47(4)/2011, 501–505.

[10] Mell P., Grance T.: The NIST Definition of Cloud Computing (Technical report). National Institute of Standards and Technology: U.S. Department of Commerce, 2011 [DOI:10.6028/NIST.SP.800-145].

[11] Neff M., Kipp M., Albrecht I., Seidel H.P.: Gesture Modeling and Animation by Imitation. MPI–I 4/2006.

[12] Ong E.I., et al. : Sign language recognition using sequential pattern trees. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, 2200–2207.

[13] Raheja J.: Android based portable hand sign recognition system. 2015 [DOI: 10.15579/gcsr.vol3.ch1].

[14] Shapiro A., Chu D., Allen B., Faloutsos P.: Dynamic Controller Toolkit, 2005 http://www.arishapiro.com/Sandbox07_DynamicToolkit.pdf (available 15.05.2019).

[15] Smith J., Navi R.: The Architecture of Virtual Machines. Computer. IEEE Computer Society 38(5)/2005, 32–38.

[16] Tensorflow framework documentation https://www.tensorflow.org/api/ (available 15.05.2019).

[17] The Linux Information Project, Cross-platform Definition.

[18] Unity3D framework https://unity3d.com/ (available 15.05.2019).

[19] YAML – The Official YAML Web Site http://yaml.org/ (available 15.05.2019).

M.Sc. Serhii Kondratiuk
e-mail: kondratiuk@univ.net.ua

In 2013 graduated from the faculty of cybernetics KNU of Taras Shevchenko. In 2015-2019 studied as a Ph.D. student at faculty of computer science and cybernetics. Since 2017 works as an assistant of cathedral of Theoretical Cybernetics.

ORCID ID: 0000-0002-5048-2576

Prof. Iurii Krak
e-mail: krak@univ.kiev.ua

In 1980 graduated from the Faculty of Cybernetics Taras Shevchenko National University of Kyiv (KNU), in 1984 – Post Doctorate, in 1999 - Doctorate of KNU, 1998 - Internship at the Yale University (USA). 1989 – Assistant Professor, 1992 - Associate Professor, 1999 – Full Professor, 2014 - Head of Theoretical Cybernetics Department of KNU, 2018 – Corresponding Member of NAS of Ukraine.

ORCID ID: 0000-0002-8043-0785

Prof. Waldemar Wójcik
e-mail: waldemar.wojcik@pollub.pl

Director of Institute of Electronic and Information Technologies, Faculty Electrical Engineering and Computer Science, Lublin University of Technology. His research interests include electronics, automatics, advanced control techniques, the optimization of the industrial processes, and fiber optic sensors including fiber Bragg gratings.

ORCID ID: 000-0002-0843-8053

otrzymano/received: 15.05.2019 *przyjęto do druku/accepted: 15.06.2019*