Fingerspelling Alphabet 3D Modeling and Recognition Base on CNN Technology for Cross Platform Applications

Serhii Kondratiuk[1[0000-0002-5048-2576]], Iurii Krak[1,2[0000-0002-8043-0785]], Olexander Barmak[3[0000-0003-0739-9678]], Anatolii Pashko[1[0000-0001-6944-8477]]

[1]Taras Shevchenko National University of Kyiv, 64/13 Volodymyrska str., 01601, Ukraine
{sergey.kondrat1990,aapashko}@gmail.com, krak@univ.kiev.ua
[2]Glushkov Cybernetics Institute, Kyiv, 40 Glushkov avenue, 03187, Ukraine
krak@nas.gov.ua
[3]National University of Khmelnytsky, 11, Institutes str., 29016, Ukraine
barmakov@khnu.km.ua

Abstract. The technology, which is implemented with cross platform tools, is proposed for modeling of gesture units (fingerspelling alphabet) of sign language, animation between states of gesture units with a combination of gestures (words). Implemented technology simulates sequence of gestures using virtual spatial hand model and performs recognition of dactyl items from camera input. With the cross platform means technology achieves the ability to run on multiple platforms without re-implementing for each platform.

Keywords: cross platform, sing language, modeling, recognition, convolutional neural network.

1 Introduction

Modern hardware is able to collect information fast and almost without restriction, process data both in cloud computing (model, which provides a universal, easy access on demand through the network to the virtual cluster computing resources) [1] and locally on the device, and through data channel processing results are returned to the user. All this is also true for sign language [2], [3]. Sings can be stored and reproduced via a variety of devices and platforms, stationary or mobile, high performance or energy efficient. The actual problem is the reproduction of sign language on all these platforms, for further usage by people with hearing disabilities in particular and everyone in general. Deployment of a single unified technology on various platforms (android, ios, windows, linux, web) without need to port it or to implement it under each platform is a major problem.

One way of solving the stated problem of visualization and reproduction of sign language is cross platform software development. Unlike single-platform technologies

that operate only on a specific platform under which they were developed, “cross platform software provides the ability to perform on more than one platform with identical (or nearly identical) functionality” [4]. The term “platform” in this context may refer to one of or a combination of several definitions: 1) the type of operating system (such as Microsoft Windows, Mac OS X, Linux, Solaris, Android, iOS); 2) processor type (such as x86, PowerPC, ARM); 3) the type of hardware (e.g., mainframe, workstation, personal computer, mobile device) [4]. Cross-platform technologies are on a par with the platform independent technologies (those that can operate on any platform, such as Web application) [4] and cross-platform virtual machines (technologies that support individual processes or systems, depending on the level of abstraction at which is virtualization) [5].

In this article the proposed solution of the problem is via cross platform development, taking into account characteristics of different classes of devices (such as hardware, CPU power, amount of memory, presence on the Internet) and setting the number of polygons of the three-dimensional hand model and gesture animation step. Given paper is a progress of author's investigations [2], [6], [7]. Gesture modeling and gesture recognition is performed via cross platform means as a part of proposed communication technology.

2 Approaches for modeling and recognition of sign language and problem statement

Sign modeling is a problem that is considered both independently and as part of the problem of modeling and recognition of gestures and thus as a technology learning and evaluating sign language. One of the systems to display the sign language is American Sing Language Online Dictionary [8], which consists of a video database of words and phrases displayed via sign language. These developments were involved in a number of commercial agencies [7], but the systems they propose are configured to pre-determined number of gestures, and therefore do not solve the problem of modeling sign language. Also all of them lack functionality of gestures recognitions, thus not allowing to evaluate quality of sing language performed by a user.

Creating a model hand is the first step in the task of sing language modeling. In their work [10], authors analyze existing approaches of hand modeling, which are divided into two main groups: spatial and temporal. Former consider the characteristics of different positions for the hand gestures, while latter refer to the description of the dynamics of gestures. Modeling hands in the spatial area can be completed in two and three dimensions.

In [11] proposed system by authors is able to simulate sign animations for a given text. As a part of this system a statistical model is used to analyze input text and generative algorithm is used when creating the appropriate simulated kinematics of sign animations. Within the article, the authors have provided ANVIL tools for input text annotation, gesture generator NOVA, and DANCE library developed in [12] is used for gesture animation. The system is built on the Microsoft Windows platform and

x86 processor. In [11]-[18] authors discuss the modeling of virtual character for spatial reproduction of sign language. The training system is based on Microsoft Windows platform and x86 processor. Gesture recognition for mobile platforms is developed in [19], but gesture modeling on mobile devices is not performed.

The proposed technology should perform modeling of sign units [13], [14] of sign language, and reproduce animation of gestures structures (words, sentences) via state transitions between shown units using spatial virtual model hand. The proposed technology should perform recognition of sign language based on camera input from the device in order to evaluate sing language performed by user . The technology should be a combined solution for learning sing language via gesture modeling and recognition.

Technology should solve the problem of running on existing platforms using cross platform development without implementing the functionality for each platform separately. The effectiveness of the proposed approach is shown in building cross platform technology for modeling and recognition of Ukrainian dactyl (fingerspelling) alphabet.

3 Proposed methods for gestures modeling

To address the modeling of sign language and perform animation of sign structures using spatial virtual model hand the cross platform technology based on cross platform framework Unity3D [20] is proposed. Cross platform framework Unity3D is also used for the user interface, both libraries and technology are implemented with programming language C#. Proposed tools can solve the problem of running the technology on multiple existing platforms. The novelty of the proposed technology is that it is cross platform and has customizable level of polygons for three dimensional hand model and animation step for gesture transitions. This allows to run proposed technology without changes on multiple platforms (different types of processors, operating systems and hardware).

Advantage of cross platform technology over technologies developed for a single platform is that there is no need to modify or re-implement the functionality already available for other platforms (porting) [4], which speeds up the process of developing and deploying technologies, and increases the number of potential users. The advantage of cross platform technology over cross platform virtual machine emulators is performance speed and absence of necessity to install additional software (software dependencies).

The core of the technology is composition of three cross-platform modules (Fig. 1): three dimensional hand model (which is implemented with cross platform framework Unity3D), user interface (implemented also with cross platform framework Unity3D) and gesture recognition module (implemented with cross platform framework Tensorflow [21]. Core functionality is implemented with C# and Python and runs on desktop OS (MacOS, Linux, Windows) and on mobile OS (Android, iOS).

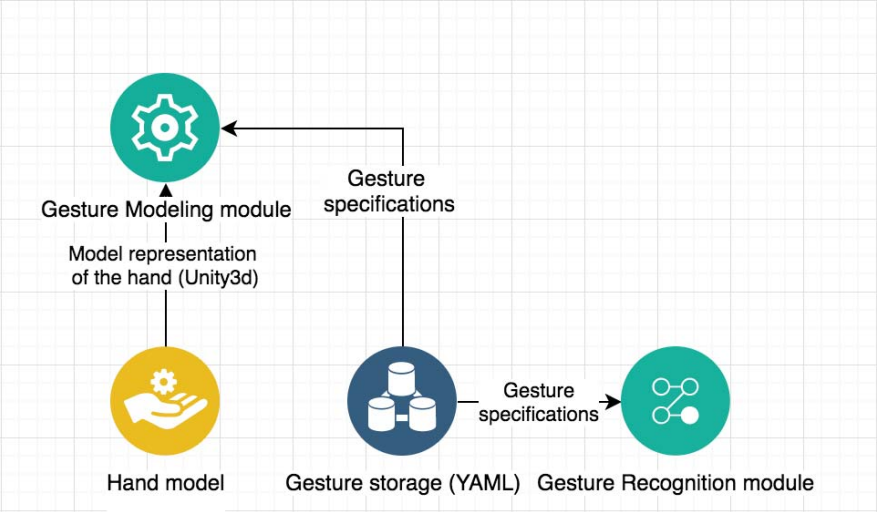

Fig. 1. Diagram of infological model of cross platform gesture communication technology

Hand model module is cross platform and provides hand model representation for gesture recognition module. Hand renderer receives hand model representation and gesture specifications from gesture storage module, and provides a high-polygon rendered hand model. Gesture learning module and gesture modification module are implemented with cross platform Unity3D, both taking as input results of hand model renderer. Gesture modification module provides updated gesture specifications and transmits them to gesture storage. Gesture recognition module is proposed to be implemented with Tensorflow framework and receives as input hand model, gesture specifications and input from camera

The hand model which is built in gesture modeling module has 27 bones, 8 of the bones are in wrist, 3 are in the thumb (one metacarpal and 2 phalanx) and 4 metacarpus and 12 phalanges are in other fingers. Each bone is connected to the other through different types of joints.

Designing your own cross platform engine for simulating the hand is non-trivial task, thus as the core technology for modeling three-dimensional hand model and gesture animations between morphemes cross platform framework Unity3D was selected. Unity3D framework is able to effectively reproduce a realistic hand model which consists of more than 70,000 polygons (Fig. 2).

Based on the anatomy of the hand within Unity3D hand model was developed with 25 degrees of mobility, four of them located in the metacarpal-carpal joint, to the little finger and thumb to provide movement palm. The thumb has 5 degrees of freedom, middle and index fingers have 4 degrees of freedom (metatarsophalangeal joint with two degrees of mobility, and the distal and proximal interphalangeal joints each have one). To preserve the gesture YAML format was selected [22]

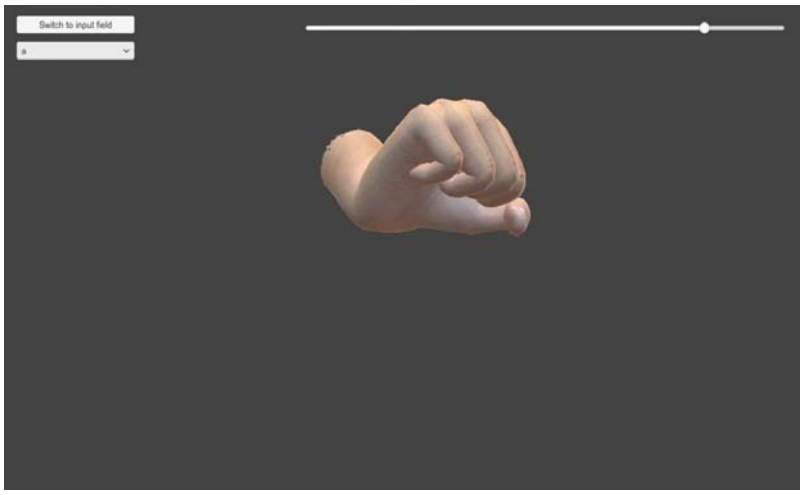

Fig. 2. 3D model of gestures under iOs platform

4 Proposed methods of gestures 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. For example, Ong et al. propose a novel method called Sequential Pattern Mining (SPM) that utilizes tree structures to classify signs [23].

Convolutional Neural Networks (CNNs) 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 sing languages show that CNNs achieve state-of-the-art accuracy for gesture recognition [24],[25].

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.

For experiment there was collected a dataset with Ukrainian dactyl language letters (fingerspelling alphabet). Each gesture consists of 1000 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.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Fig. 3. MobileNet body architecture [26]

MobileNet[26] architecture (see Fig. 3) was used as a 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.

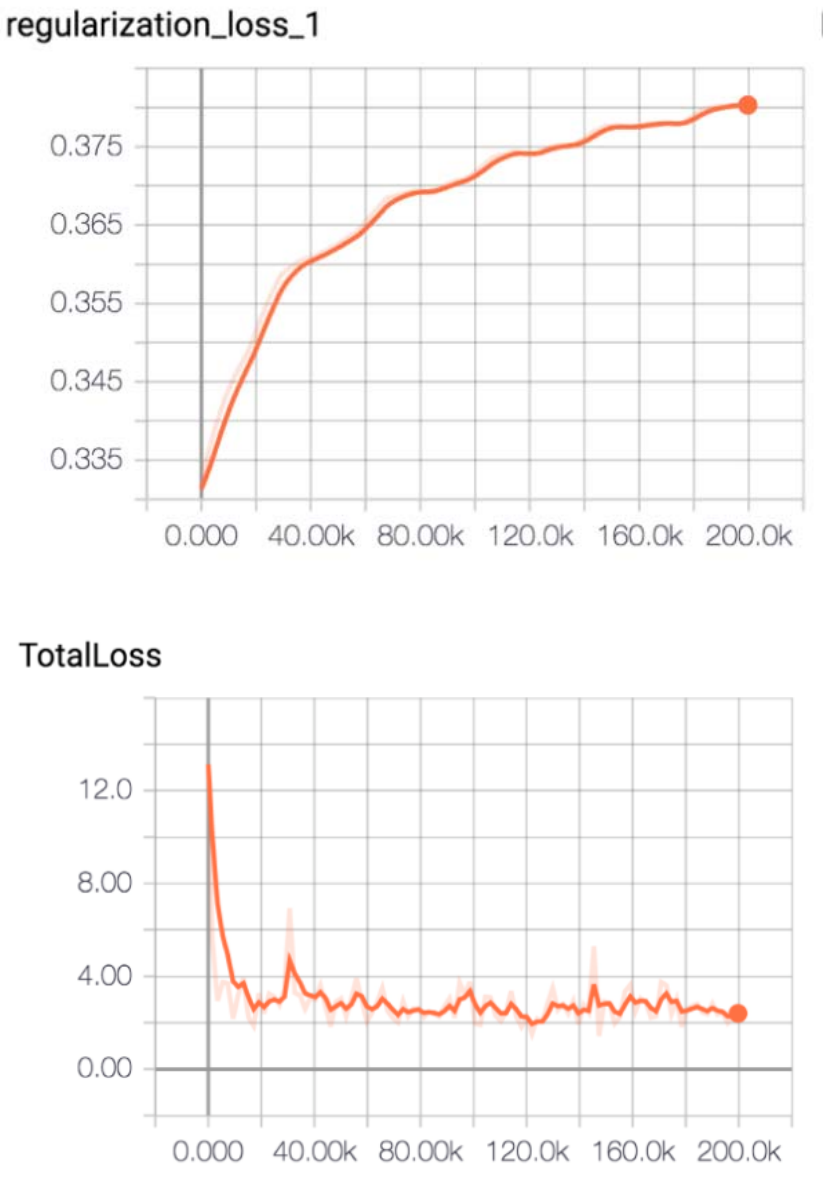

Fig. 4. Curves of the neural network optimization

Process of training MobileNet network for gesture detection takes ~ 200.000 iterations, which is approximately 10 epochs. On the Fig. 4 shows curves of how the neural network is optimized.

On the Fig. 5 shows example of how the proposed technology detects specific gesture from Ukrainian dactyl and draws a bounding box over a detected gesture.

Fig. 5. Example of fingerspelling letter (b)

5 Application of cross platform tools

Due to selected cross platform implementation tools, the proposed technology solves the problem of executing on multiple platforms without the implementation under each platform separately.

Software offered and used in the implementation of information technology is cross-platform and operates unchanged regardless of operating system (Windows, Linux, Android, iOS), CPU type (x86, arm), and the type of hardware (mobile or stationary device).

With its cross platform build system Unity3D it is possible to create applications for each platform without porting or changing the original code.

As there are no specific hardware requirements for information technology for modeling sign language, there are objective obstacles for performance speed of older generations devices. To overcome this problem, the following adaptive approach to information technology was proposed[6].

Further modules implementation will leverage from existing cross-platform technology. Gesture learning and gesture recognition modules, developed with cross platform technologies (Python, Tensorflow) will be embedded into information and ges-

ture communication cross-platform technology. In case of the mobile app (iOS, Android) or application on the device with a stationary operating system (Windows, Linux), during installation on the device, information technology analyzes the existing hardware and, depending on its capacity, conducts a series of adjustments: 1) number of polygons of the hand model changes to priority for performance speed; 2) during rotation hand model changes pitch angle at which it rotates, with priority for speed. If the available hardware does not meet the minimum requirements of information technology, the user is given the recommendation to choose "online" mode, in which the calculation is not performed on hardware.

6 Conclusion

The proposed technology is built with cross platform tools for gesture modeling, gesture transitions animation and gesture recognition. The technology uses virtual spatial model of hand. With the help of cross platform development, the technology solves the problem of execution on the existing multiple platforms without implementing functionality under each platform separately. Thus, it was shown the effectiveness of the technologies built using cross platform tools, for example modeling and recognition elements of dactyl Ukrainian alphabet sign language. Information and gesture communication technology was developed with further scaling capabilities in mind for gestures of other languages alphabets.

To implement this idea, the validation mechanism of new gestures to the common database can be applied. Cross platform information and communication technology and standardized protocol and data format (YAML) allows a range of solutions for remote computing using cloud computing, Web servers, local servers using a single sign database PostgreSQL [27]. The gesture communication technology can be augmented with other cross platform modules, such as gesture recognition and gesture learning modules.

References

1. Mell, P., Grance, T.: The NIST Definition of Cloud Computing (Technical report). National Institute of Standards and Technology: U.S. Department of Commerce. Special publication 800-145 (2011) doi:10.6028/NIST.SP.800-145.
2. Kryvonos, I.G., Krak, I.V., Barmak, O.V., Kulias, A.I.: Methods to Create Systems for the Analysis and Synthesis of Communicative Information. Cybernetics and Systems Analysis 53(6): 847-856 (2017)
3. Stein, D., Dreuw, P., Ney, H., Morrissey, S., Way A.: Hand in hand: automatic sign language to English translation. In: Proceeding of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2007), Skovde, Sweden, 7-9 September 2007, pp. 214-220 (2007)
4. The Linux Information Project, Cross-platform Definition [http://www.linfo.org/cross-platform.html] (2005)
5. Smith, J., Nair, R.: The Architecture of Virtual Machines. Computer 38(5): 32-38 (2005)

6. Krak, I., Kondratiuk, S.: Cross-platform software for the development of sign communication system: Dactyl language modelling. In: Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017, Lviv, 05-08 September 2017, vol. 1, pp. 167-170 (2017)
7. Kryvonos, Iu.G., Krak, Yu.V., Barchukova, Yu.V., Trotsenko, B.A.: Human hand motion parametrization for dactylemes modeling. Journal of Automation and Information Sciences 43 (12): 1-11 (2011)
8. ASL Sing language dictionary [http://www.signasl.org/sign/model] (1999)
9. 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] (2014)
10. Rafiqul, Z. K., Noor, A. I., Natarajan, M., et al : Comparative study of hand gesture recognition system: SIPM, FCST, ITCA, WSE, ACSIT, CS & IT 06, pp. 203-213 (2012)
11. Neff, M., Kipp, M., Albrecht, I., Seidel, H.-P.: Gesture Modeling and Animation by Imitation. MPI–I–2006–4-008 (2006)
12. Shapiro, A., Chu, D., Allen, B., Faloutsos, P.: Dynamic Controller Toolkit [http://www.arishapiro. com/Sandbox07_DynamicToolkit.pdf] (2005)
13. Kryvonos, Iu.G., Krak, Iu.V.: Modeling human hand movements, facial expressions, and articulation to synthesize and visualize gesture information. Cybernetics and Systems Analysis 47(4): 501-505 (2011)
14. Kryvonos, Iu.G., Krak, Iu.V., Barmak, O.V., Shkilniuk, D.V.: Construction and identification of elements of sign communication. Cybernetics and Systems Analysis 49(2): 163-172 (2013)
15. The 3D Biomechanics Data Standard [https://www.c3d.org/]
16. Holte, M.B., Moeslund, T.B.: Gesture recognition using a range camera, Technical Report. CVMT-07-01 (2007)
17. Krak, Yu.V., Golik, A.A., Kasianiuk, V.S.: Recognition of dactylemes of Ukrainian sign language based on the geometric characteristics of hand contours defects. Journal of Automation and Information Sciences 48(4): 90-98 (2016)
18. Krak, I.V., Kryvonos, I.G., Barmak, O.V., Ternov, A.S.: An Approach to the Determination of Efficient Features and Synthesis of an Optimal Band-Separating Classifier of Dactyl Elements of Sign Language. Cybernetics and Systems Analysis 52(2): 173-180 (2016)
19. Raheja, J. I., Sadab, A.S., Chaudhary, A.: Android based portable hand sign recognition system, Science Gate Publishing, USA, pp. 1-18 (2015) doi: 10.15579/gcsr.vol3.ch1
20. Unity3D framework [https://unity3d.com/]
21. Tensorflow framework documentation [https://www.tensorflow.org/api/]
22. YAML - The Official YAML Web Site [http://yaml.org/]
23. Ong, E.-J., et al.: Sign language recognition using sequential pattern trees. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, 16-21 June 2012, pp. 2200-2207 (2012) doi**:** 10.1109/CVPR.2012.6247928
24. Garcia, B.: American Sign language: Real-time American Sign Language Recognition with Convolutional Neural Networks. Stanford University, Stanford, CA (2015)
25. Bobic, V., Tadic, T., Kvascev, G.: Hand gesture recognition using neural network based techniques, In: NEUREL 2016: 2016 13th Symp. on Neural Networks and Applications (NEUREL), Belgrade, 22-24 November 2016, pp.35-38 (2016)
26. Howard, A.G., Zhu, M., Bo C., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, [https://arxiv.org/pdf/1704.04861.pdf] (2017)
27. PostgreSQL official web site [https://www.postgresql.org/]