УДК 004.93

С.С. Кондратюк
Київський національний університет імені Тараса Шевченка, Україна
вул. Володимирська, 60, м. Київ, 01601

РОЗПІЗНАВАННЯ ЖЕСТІВ УКРАЇНСЬКОЇ ДАКТИЛЬНОЇ АБЕТКИ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ІЗ ТРИВИМІРНОЮ ЗГОРТКОЮ

S. Kondratiuk
Taras Shevchenko National University of Kyiv, Ukraine
60, Volodymyrska St., Kyiv, 01601

UKRAINIAN DACTYL ALPHABET GESTURE RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS WITH 3D CONVOLUTIONS

Запропонована технологія, розроблена за допомогою кросплатформених засобів, для моделювання жестів української дактильної абетки, анімації переходів між станами жестових одиниць та комбінування жестів (слів).Технологія відтворює послідовність жестів за допомогою віртуальної просторової моделі руки та виконує розпізнавання дактилем із вхідного потоку камери за допомогою навченої на зібраному наборі зображень згорткової нейронної мережі, із взятою за основу архітектурою MobileNetv3, та з підібраною оптимальною конфігурацією шарів та параметрів мережі. На зібраному тестувальному наборі даних досягнуто точності у понад 98%.

Ключові слова: кросплатформеність, мова жестів, моделювання дактилем, розпізнавання дактилем, згорткові нейронні мережі, mobilenet

The technology, which is implemented with cross platform tools, is proposed for modeling of gesture units 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 using trained on collected training dataset set convolutional neural network, based on the MobileNetv3 architecture, and with the optimal configuration of layers and network parameters. On the collected test dataset accuracy of over 98% is achieved.

Keywords: cross platform, sing language, dactyl modeling, dactyl recognition, convolutional neural net-works, mobilenet

Introduction

Gesture based communication is one of real methods for data transition, close by with content and discourse. Sings 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 [1]) and others don't, and so forth. Displaying such a technology for sing 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 sing language.

Cross-platform development [2] give an approach to beat this issue. Cross-platform development can be utilized instead of virtual-machines [3] 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 sing 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 sings progress) in view of the CPU type, measure of accessible memory and web connection speed. The sing recognition is additionally performed utilizing cross-platform developments and can be altered for the trade-off in model size and execution speed. The sing (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 [4], [5].

Existing approaches for recognition of sign language

Detection of hand gestures can be considered as a type of task of object detection, which has a set of mature and novel approaches in both classic computer vision and deep learning, with convolutions neural networks specifically.

Since release of the convolutional neural network architecture for ImageNet contest, AlexNet [6], this new approach proved to show robust results in different condition of input data. Neural networks show robust recognition quality for object detections, when object have diverse distortions, different scale and various light conditions, noise, blur.

One of the key idea in transferring from static object detection into dynamic object detection is to use multiple subsequent frames from the video input instead of a single image, in order to utilize additional temporal data among with spatial data.

As bigger datasets with recorded activities were released (Sports-1M [7], Kinetics [8], Jester [9]), convolutional neural networks with 3-dimensional convolutions because successful. The size of the dataset allowed to train the model without overfitting [10].

Gestures of sign language were detected using different approaches based on classic computer vision with hand-crafted features such as orientation of histograms [11], histogram of oriented gradients (HOG) [12] or bag-of-features [13]. Although the state of the art hand gesture recognition architectures are based on CNNs [14, 15, 16], similar to other computer vision tasks.

Commonly in order to achieve higher performance in terms of accuracy on the gesture dataset, the architecture of the CNN was made more complex [17, 18].

However, the proposed technology in this paper aims to work cross-platform, an a various set of platforms and devices, some of them, such as hand-held devices (smartphones and tables) have limited computational resources and capabilities. Thus, research of existing approaches among CNN was accented on lightweight architectures which show satisfying performance on mobile cpus, such as SqueezeNet [19], MobileNet [20], MobileNetV2 [21], ShuffleNet [22] and ShuffleNetV2 [23], MobileNetV3 [24] which aim to reduce computational cost but still keep the accuracy high. In our work, we have used the 2D and 3D versions of MobileNetV3.

Problem statement

The proposed technology should consist of two parts, which are sign language [25] modeling and gesture recognition module. Both modules should be able to run without codebase modification on multiple platforms and should be developed using cross-platform tools.

Gesture recognition module should consist of a model which is able to detect and identify the gesture, specified by the user, from a camera input. Set of gestures is limited by the Ukrainian dactyl language, but can be extended further. An appropriate dataset of Ukrainian dactyl language should be collected for testing the model performance. The sing language modeling module should be able to reproduce a gesture specified by a set of parameters, stored in a database, and should be limited by a set of Ukrainian dactyl language signs, but can be extended further with other languages.

The gesture recognition module should utilize the model which show robust and state-of-the-art performance along with high efficiency in terms of computational resources in order to achieve high accuracy and FPS-rate on various platforms, using cross-platform technologies.

Proposed approach

To developed a technology for Ukrainian dactyl language modeling and recognition, which can run on multiple platforms, without changing the codebase, an approach based on cross-platform tools is proposed. Gesture modeling module should consist of a virtual three dimensional hand model and a user interface, which should provide the user with ability to specify a symbol or a set of symbols, which then will be transitioned as a sequence of gestures. To implement both hand model and user interface, a cross-platform framework Unity3D [26] was used. Comparing to other 3D engines, it provides a unified development process for all available platforms (mobile, desktop and web) and provides a seamless way to deploy the application on all of them without changing the codebase. To develop a gesture recognition module, a cross-platform framework Tensorflow [27] is proposed. This approach based on cross-platform framework for machine learning allows to developed and train a gesture recognition model once, and then deploy it on multiple platforms (mobile, desktop and web) without any modifications to the model or the code for training. As a model architecture, the MobileNet architecture is considered, enhanced with 3D convolutions, to take into account temporal information from a sequence of input frames from the camera. Altogether, the proposed technology novelty is that it's a unified cross-platform technology for Ukrainian dactyl language modeling and recognition, with improved MobileNet architectture for improved recognition of the Ukrainian dactyl alphabet.

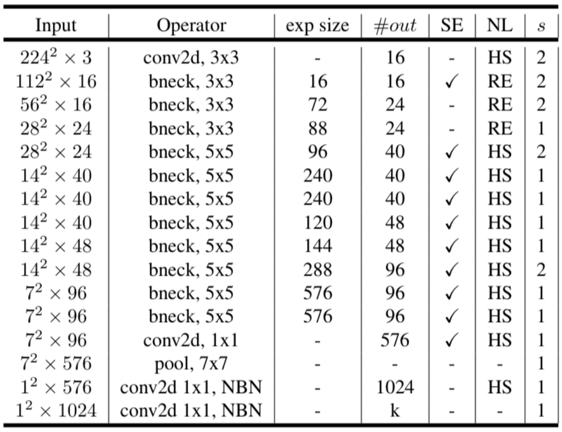
Gesture recognition

Gesture recognition, as a part of cross-platform technology for Ukrainian dactyl language modeling and recognition, should be implemented using cross-platform tools. Gesture recognition approach depend on the type of input information they work with. In case of 3D model bases algorithms or skeletal-based algorithms, the approach can use volumetric or skeletal model, or a combination of them. Although, these approaches tend to be computationally expensive and require additional hardware from user. Other type of approaches, appearance-based models derive parameters directly from the image or a sequence of images (in case video is used as an input). As a next step some pattern mining technique or machine learning approach is used to train a recognition model. Due to no need in additional hardware apart from a simple webcamera, these type of approaches were selected for the cross-platform technology. Some approaches, for example, Ong et al. [28] proposes Sequential Pattern Mining in order to detect signs based on the tree structures.

Convolutional Neural Networks (CNN) is a class of deep neural networks which are regularized versions of multilayer perceptrons, most commonly applied to analyzing images and videos. CNNs are especially good at analyzing images due to ability to take into account locality reference of the data in the image (typically nearby samples at some input data are not related, which is not true in case of an image). Therefore, CNN show state-of-the-art results in image classification and recognition tasks [29], [30]. Another benefit of the convolutional neural networks is no need in hand-crafted features, unlike conventional pattern matching algorithms. The process of training takes the input data and finds all the features needed for recognition and stores them as weights of the model. CNNs are robust at the task of classification or recognition of the object on an image, independent of input image scale, lightning conditions, occlusions, noise, etc. Although training such a model requires a sufficient dataset. Typically architecture of the CNN consists of a set of convolutional, pooling and ReLU layers. Tensorflow framework provides a cross-platform and performance-efficient implementation of convolutional neural networks.

Gesture recognition with MobileNet

MobileNetV3 architecture (Figure 1) is a new mobile architecture, development of the MobileNet model. MobileNetV3 extends its predecessor with 2 main ideas. Residual blocks connect the beginning and end of a convolutional block with a skip connection. By adding these two states the network has the opportunity of accessing earlier activations that weren't modified in the convolutional block. This approach turned out to be essential in order to build networks of great depth. On the other hand, MobileNetV3 follows a narrow->wide->narrow approach. The first step widens the network using a 1x1 convolution because the following 3x3 depthwise convolution already greatly reduces the number of parameters. Afterwards another 1x1 convolution squeezes the network in order to match the initial number of channels. A factor of 6 opposed to the 4 in our example. *c* represents the number of input channels and *n* how often the block is repeated. Lastly, *s* tells whether the first repetition of a block used a stride of 2 for the downsampling process. This is a common assembly of convolutional blocks.



Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

Fig. 1. Architecture of MobileNetv3.

Network Improvements have been made in two ways: Layer removal (1) and swish non-linearity (2).

In the last block, the 1x1 expansion layer taken from the Inverted Residual Unit from MobileNetV2 is moved past the pooling layer. This means the 1x1 layer works on feature maps of size 1x1 instead of 7x7 making it efficient in terms of computation and latency.

We know that the expansion layer takes a lot of computation. But now that it is moved behind a pooling layer, we don’t need to do the compression done by projection layer from the last layer from the previous block. Thus we can remove that projection layer and the filtering layer from the previous bottleneck layer (block).

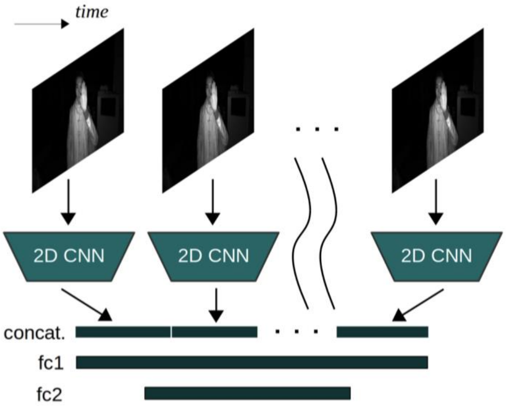


Fig. 2. Pipeline of 3D-convolutions with MobileNetv3.

Ukrainian dactyl alphabet dataset collections for recognition with MobileNet

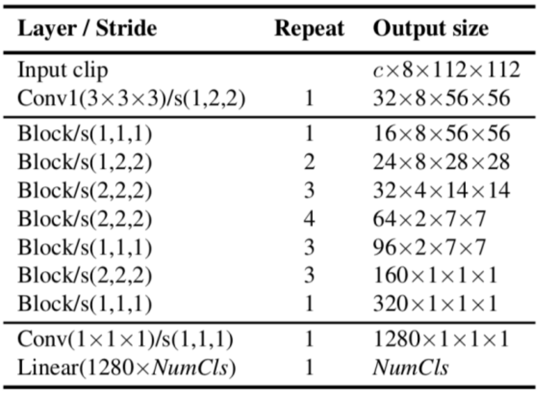
Since training of the Convolution Neural Network hardly depends on a big and diverse dataset, to achieve a high enough accuracy metrics level, dataset of Ukrainian dactyl language letters with diverse characteristics was collected. Each gesture consists of 1500 sample images, and 50 different people 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. Overall ~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.

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.

MobileNetv3 with 3D convolutions

Figure 2 shows the pipeline with spatiotemporal modeling approach used for 2D CNN models. Features of each 8 frames are extracted using the same 2D CNN and concatenated keeping their order intact. Afterwards, two levels of fully connected (fc) layers are applied in order to get class-conditional probability scores. The reason behind is that fc layers can organically infer the temporal relations, without knowing it is a sequence at all. The size of features 2D CNNs extracts is 64 for each frame. With the first fc layer, feature dimension is reduced from 64×8=512 to 256. With the second fc layer, dimension is reduced to the number of classes.



Layer / Stride	Repeat	Output size
Input clip		$c\times8\times112\times112$
Conv1(3×3×3)/s(1,2,2)	1	32×8×56×56
Block/s(1,1,1)	1	16×8×56×56
Block/s(1,2,2)	2	24×8×28×28
Block/s(2,2,2)	3	32×4×14×14
Block/s(2,2,2)	4	64×2×7×7
Block/s(1,1,1)	3	96×2×7×7
Block/s(2,2,2)	3	160×1×1×1
Block/s(1,1,1)	1	320×1×1×1
Conv(1×1×1)/s(1,1,1)	1	1280×1×1×1
Linear(1280×$NumCls$)	1	$NumCls$

Fig. 3. Architecture of MobileNetv3 with 3D-convolutions.

On the other hand, 3D CNNs contains spatiotemporal modeling intrinsically and does not require an extra mechanism. We have inflated SqueezeNet and MobileNetV3 such that they accept 8 frames as input. The details of the 3D-MobileNetV3 are given in Figure 3.

Gesture recognition experiment

Standard techniques of fighting overfitting of the neural network were applied on each training.

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, so the obtained architecture was decided as optimal in terms of the smallest architecture with best accuracy which is shown in Figure 4 (macro average f1-score and confusion matrix).

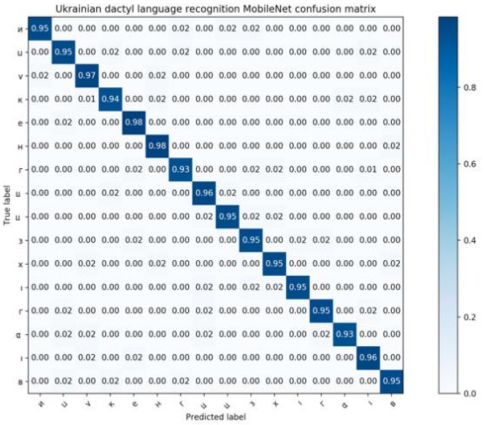


Fig. 4. Confusion matrix of the optimal architecture model.

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 [31] 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 tradeoff 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. Mell, P., Grance, T. (2011, September). The NIST Definition of Cloud Computing (Technical report). National Institute of Standards and Technology: U.S. Department of Commerce. doi:10.6028/NIST.SP.800-145. Special publication 800-145.
2. The Linux Information Project, Cross-platform Definition.
3. Smith, J., Nair, R. (2005). The Architecture of Virtual Machines. Computer. IEEE Computer Society. 38 (5): 32–38.
4. Krak, I., Kondratiuk, S. (2017). 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, 167-170. DOI: 10.1109/STC-CSIT.2017.8098760
5. Krak, Y.G., Krak, Y.V., Barchukova, B.A. (2011). Human hand motion parametrization for dactylemes modeling, *Journal of Automation and Information Sciences*, 43 (12), 1-11.
6. Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, 1097-1105.
7. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. *In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1725-1732.
8. Zisserman, A., Carriera, J. (2017). Action recognition a new model and the kinetics dataset. In Computer Vision and Pattern Recognition (CVPR). IEEE Conference on pages 4724-4733, IEEE.
9. Materzynska, J., Berger, G., Bax, I., Memisevic, R. (2019). The Jester Dataset: A Large-Scale Video Dataset of Human Gestures.
10. Haha, K., Kataoka, H., Satoh, Y. (2018). Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognitions*, Salt Lake City, UT, USA, 18-22.
11. Freeman, W., Roth, M. (1995). Orientation histograms for hand gesture recognition. In International workshop on automatic face and gesture recognition, vol. 12, 296-301.
12. Prasuhn, L., Oyamada, Y., Mochizuki, Y., Ishikawa, H. (2014). A HOG-based had gesture recognition system on a mobile device. In 2014 IEEE *International Conference on Image Processing (ICIP)*, 3973-3977, IEEE.
13. Dardas, N., Georganas, D. (2011). Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Transactions on instrumentation and measurement, 60: 3592-3607.
14. Kopuklu, O., Kose, N., Rigoll, G. (2018). Motion fused frames: Data level fusion strategy for hand gesture recognition arXiv preprint arXiv:1804.07187
15. Molchanov, P., Gupta, S., Kim, K., Pulli. K. (2015). Multi-sensor system for driver’s hand-gesture recognition. In Automatic Face and Gesture Recognition (FG), 11th IEEE Inter- national Conference and Workshops on, vol. 1, 1–8. IEEE
16. Molchanov, P., Gupta, S., Kim, K.,Kautz, J. (2015). Hand gesture recognition with 3d convolutional neural networks. *In The IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) Workshops
17. Hu, J., Shen, L., Sun, G. (2018) Squeeze-and-excitation networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.
18. He, K., Zhang, X., Ren, S., Sun., J. (2016) Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
19. Iandola, F., Han, S., Moskewicz, M., Ashraf, K., Dally, W., Keutzer, K. (2016). Squeezenet: AlexNet-level accuracy with 50x fewer parameters and 0.5 mb model size. arXiv preprint arXiv:1602.07360.
20. Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. (2017). Mobilenets: Efficient convolutional neural

networks for mobile vision applications. arXiv preprint arXiv:1704.04861

21. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In 2018 IEEE/CVF *Conference on Computer Vision and Pattern Recognition*, 4510–4520. IEEE.
22. Zhang, X., Zhou, X., Lin, M., Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In 2018 IEEE/CVF *Conference on Computer Vision and Pattern Recognition*, 6848–6856. IEEE.
23. Ma, N., Zhang, X., Zheng, H., Sun, J., (2018). Shufflenet v2: Practical guidelines for efficient CNN architecture design. arXiv preprint arXiv:1807.11164, 5.
24. Howard, A., Sandler, M., Chu, G., Chen, L., Chen, B., Tan, M., Wang, W. (2019). Searching for MobileNetV3. axXiv:1905.02244, 5
25. ASL Sing language dictionary. URL: http://www.signasl.org/sign/model
26. Unity3D framework. URL: https://unity3d.com/
27. Tensorflow framework documentation. URL: https://www.tensorflow.org/api/
28. Ong, E. (2012). Sign language recognition using sequential pattern trees. In: *Computer Vision and Pattern Recognition* (CVPR), IEEE Conference on IEEE pp. 2200-2207.
29. American Sign language: Real-time American Sign Language Recognition with Convolutional Neural Networks (2015). Brandon Garcia Stanford University Stanford, CA.
30. Bobic, V. (2016). Hand gesture recognition using neural network based techniques, School of Electrical Engineering, University of Belgrade
31. PostgreSQL official web site. URL: https://www.postgresql.org/

РЕЗЮМЕ

С.С. Кондратюк

Розпізнавання жестів української дактильної абетки за допомогою згорткових нейронних мереж із тривимірною згорткою

Мова жестів є одним із основних засобів передачі інформації, поряд із текстом і мовою. Як правило, у кожної країни є своя рідна мова жестів, проте напевно, невідомо, скільки мов жестів існує у всьому світі. Українська мова жестів та український алфавіт дактилем є одними із найпоширеніших засобів спілкування в Україні після текстового та розмовного спілкування.

Надання технології вивчення жестів (знаків, дактилем) української мови жестів для такої спільноти є актуальною проблемою та складним завданням.

Для вирішення задачі моделювання мови жестів та виконання анімації жестових структур за допомогою просторової віртуальної моделі руки пропонується кросплатформна технологія, заснована на кросплатформеній бібліотеці Unity3D. Кросплатформена бібліотека Unity3D також використовується для інтерфейсу користувача, технологія реалізована за допомогою мови програмування C#. Запропоновані інструменти можуть вирішити проблему запуску технології на декількох існуючих платформах. Новизна запропонованої технології полягає в тому, що вона є кросплатформеною та має настроюваний рівень полігонів для тривимірної моделі руки та крок анімації для переходів жестів.

Модель руки, вбудована в модуль моделювання жестів, має 27 кісток, кожна кістка з'єднана з іншою через різні типи суглобів. Як основну, використано технологію моделювання тривимірної моделі руки та анімації жестів між морфемами. Вона здатна ефективно відтворити реалістичну модель руки, що складається з-понад 70 000 полігонів.

Модулі розпізнавання жестів, розроблені за допомогою кросплатформених інструментів (засновані на Python, C ++), можуть бути вбудовані в інформаційну технологію. Конволюційні нейронні мережі показали надійні результати в задачах з розпізнавання із зображень та жестів. Для експерименту був зібраний набір даних з дактилемами української мови. Кожен жест складається з 1000 зразкових зображень, 50 різних людей показували жести, з розподілом 70% чоловічих та 30% жіночих рук. Були використані різні умови освітлення (з розподілом 20% зображень у поганих, 30% у посередніх та 50% при хороших умовах освітлення), 10% зображень були спотворені шумом та розмиттям.

Архітектура MobileNetv3 була використана як основа для архітектури CNN. Для покращення якості розпізнавання було використано тривимірну згортку декількох послідовних кадрів із жестами.

Її навчання тривало ~ 300 000 ітерацій, що становить приблизно 12 епох, і досягнуто ~98% точності на тестувальному наборі даних.

Надійшла до редакції 30.05.2019