UDC 004.89: 681.518

O.M. Hlazok, c.t.s., I.A. Balytska
(National Aviation University, Ukraine)

A neural network application for classification of a sign language dactylemes

The issues of creating an application to solve the problem of dactylemes classification are considered. The problem of gesture classification is solved by creating a convolutional neural network, creating a collection of images for training, and applying computer vision functions. The proposed software may be used in the input subsystem of information and control systems. In particular, using such an application will facilitate communication with people with special needs.

Nowadays, the communication of people with disabilities is still an actual problem. According to the World Health Organization, about 460 million people in the world suffer from hearing loss or impairment. According to the statistics [1] by the World Organization of the Deaf, the number of people who use the Ukrainian sign language is about 200 000.

The presented work is devoted to the development of human-machine interface software for computer recognition of deaf-mute alphabet signs. Deaf people use two types of sign systems – the dactyl alphabet and sign language (gestural speech). The dactyl alphabet is a system of hand signs corresponding to letters, also referred to as dactylemes.

The sets of symbols that make up dactyl alphabets differ for different languages. In some countries (e.g. in Great Britain), the symbols of the dactyl alphabet are shown with two hands. The Ukrainian dactyl alphabet suggests using one hand. Ukrainian dactylology is a system of one-handed finger signs (dactylemes), each of which corresponds to a letter of the Ukrainian alphabet. The terms "dactyleme" and "dactylology" are generally accepted and widely used both in domestic and foreign deaf-pedagogical and scientific literature [2]. In Ukrainian dactylology, there are 33 dactylemes.

In gestural speech, gestures do not denote individual letters or sounds, but rather whole words and concepts. There are gestural languages that have developed natively in the communication of deaf people; these languages were not originally based on the oral or written way of transmitting words as a sequences of sounds, and therefore they differ in structure from verbal languages. There is also a group of tracing gestural languages that reproduce the structure of verbal speech. These are a kind of "bridge" between the language of the deaf and the language of the hearing. It is important that the dactyl alphabet is part of the gesture language.

Applications for gestural language recognition can make communication much easier for people who are deaf or hard of hearing. However, recognizing a gestural language is a complex task. For the correct translation of a gesture, you need to take into account the position and movement of the hand, facial expression and posture. A separate task is to identify the boundaries of the gesture in time (beginning and end). Dactylemes classification is no less important, but an easier task.

The simplest input device can be an ordinary web camera. The image obtained from it can be further processed to obtain information about the position of the user's hands.

One should also mention specialized hardware and software solutions applicable for input subsystems:

1. Sony Depthsensing Solutions, formerly a Belgian company named SoftKinetic, which develops gesture recognition hardware and software for real-time range imaging cameras. Their leading solutions are based on the use of time-of-flight (ToF) cameras [3]. This sensing technology allows to register the absolute position, movement and shape of environmental elements in 3D. A ToF camera measures the elapsed time (direct Time of Flight) or phase-shift (indirect Time of Flight) for the light reflected from the objects in the scene. The Time of Flight devices demonstrate a good operation range, a high field of view and sensor resolution. The ToF technology provides high distance accuracy and the obtained data are good for assured object classification.

2. Ultraleap, a company that produces infrared tracking cameras and tracking software. Ultraleap Gemini [4] is the fifth generation of the Leap Motion Service – the computer software on that processes the images. Using a computer vision model for visual hand tracking, the software allows seeing the both user’s hands and starting interacting immediately, works for different hand anatomies, various lighting conditions, and compatible for different platforms and camera hardware.

Next to the task of images obtaining, there is a problem of image classification. Image classification is a process performed with the aim to assign a certain class to an image [5]. As a rule, image classification refers only to images in which only one object is located and analyzed.

Convolutional artificial neural networks are often used for image classification. A convolutional neural network implements a deep learning algorithm that can take an image as input, assign importance (weights and offsets) to different aspects (objects) of an image, and be able to distinguish one objects from another. The architecture of a convolutional neural network is similar to the structure of the connection of neurons in the human brain and was inspired by the organization of the visual cortex [6].

The architecture of a convolutional neural network usually consists of convolutional, aggregation (pooling) and completely connected layers. Convolutional and aggregation layers alternate among themselves.

The number of input and output channels can be specified for each convolutional layer. This is necessary in order to calculate the number of inputs and outputs of the layers [7]. The number of inputs and outputs of the convolutional layer is found according to the formulas:

$$I = W_i \times H_i \times C_i,$$

where W_i is the width of the input image; H_i is the height of the input image; C_i is the number of input channels;

$$O = W_o \times H_o \times C_o,$$

where W_o is the width of the output image; H_o is the height of the output image; C_o is the number of output channels.

The size of the image after passing through the convolutional layer is determined according to the formulas:

$$W_o = \frac{W_i - K + 2P}{S} + 1, \quad H_o = \frac{H_i - K + 2P}{S} + 1,$$

where K is the size of the convolution kernel; P is an alignment factor (by default is 0); S is the step of the convolution (by default is 1);

Convolutional neural network training is carried out on a set of data – dactylemes images that were previously divided into classes. In addition to the training data set, there should also be a test (training) set, on which the accuracy of the network classification will be checked after each training epoch. Since there are no ready-made data sets with dactylemes, the data are collected manually. The images in the set must have the same size, the same number of color channels, and the values of the inputs are reduced to some limited range (e.g. [0 ... 1]).

Together with convolutional neural networks, computer vision is used for image classification – a branch of computer technology in which the scientific foundations of object recognition are developed and application systems are created that are capable of detecting and recognizing objects by analogy with the visual perception of environment by humans. The main tasks are the detection of a specific object on a specific digital image or video recording, or confirmation of its absence, identification of characteristic features, as well as counting or quantification of homogeneous objects [8].

The Python 3 programming language was chosen for the development of the application. Python includes the following deep learning libraries: PyTorch, TensorFlow, and Keras.

PyTorch is an open source machine learning library for Python, used to develop and train neural network-based deep learning models. TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that enables researchers and developers to advance the state-of-the-art machine learning technologies, to easily build and deploy machine learning-based applications. Keras is a deep learning API written in Python that runs on the TensorFlow machine learning platform. It was designed with an emphasis on the possibility of rapid experimentation. The ability to move from an idea to a result as quickly as possible is the key to qualitative research. All three libraries are related to each other, and also have certain basic differences. Let's consider their comparative characteristics according to a number of key parameters.

1. *API level*. Keras is a high-level API capable of running on top of TensorFlow, CNTK, and Theano. It has gained favor due to its ease of use and syntactic simplicity, which facilitates rapid development. TensorFlow is a framework that provides high-level and low-level APIs. Pytorch is a low-level API focused on working directly with array expressions.

2. *Speed of work*. Keras performance is comparatively lower than Tensorflow and PyTorch.

3. *Architecture*. Keras has a simple architecture. The code using this library is easier to read and more concise. Tensorflow is not very easy to use. PyTorch has a complex architecture and lower readability compared to Keras.

4. *Debugging*. In Keras, you usually have to debug simple networks less often. But in the case of Tensorflow, debugging is quite difficult. PyTorch has better debugging capabilities compared to the other two.

5. *Data set*. Keras is generally used for small datasets as it is comparatively slower. TensorFlow and PyTorch are used for high-performance models and large datasets that require fast execution.

6. *Popularity*. With the growing demand in the field of Data Science, there has been a huge growth in deep learning technology in the industry. Thanks to this, all three libraries have gained quite a lot of popularity. Keras tops the list, followed by TensorFlow and PyTorch. It has gained immense popularity due to its simplicity compared to the other two.

So, the features of Keras are: fast prototyping; small data sets; support for multiple server systems. The features of TensorFlow are: large data sets; high productivity; functionality; detection of objects. The features of PyTorch are: flexibility; short duration of training; debugging capabilities.

Finally, the Keras library was chosen because it can work with small data sets, because the self-collected data set is not large, and because it is comparatively simpler in use.

References

1. Ukrainian sign language. Data from the World Organization of the Deaf and Dumb. – Access mode: https://www.ethnologue.com/language/ukl .
2. Garcia M.B., Revano, T.F., Cunanan-Yabut A. Hand Alphabet Recognition for Dactylology Conversion to English Print Using Streaming Video Segmentation// 2021 9th Int. Conf. on Computer and Communications Management (ICCCM): Proceedings. – 2021. – Pp. 46–51. DOI: 10.1145/3479162.3479169.
3. Time of Flight explained. – Access mode: https://www.sony-depthsensing.com/content/uploads/2020/11/time-of-flight-explained.pdf.
4. How Hand Tracking Works. – Access mode: https://www.ultraleap.com/company/news/blog/how-hand-tracking-works .
5. Richards J. Image Classification in Practice. In book: Remote Sensing Digital Image Analysis: 6th ed. – Springer, 2022. – Pp. 447-502. DOI:10.1007/978-3-030-82327-6_11.
6. Aghdam H.H., Heravi E.J.. Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification. – Springer, 2017. – 305 p.
7. Venkatesan G.R., Li B. Convolutional Neural Networks in Visual Computing: A Concise Guide (Data-Enabled Engineering). – CRC Press, 2017. – 168 p.
8. Solem J. Programming Computer Vision with Python: Tools and algorithms for analyzing images. – O'Reilly Media, 2012. – 260 p.