CYBERNETICS and COMPUTER TECHNOLOGIES

UDC 004.852 DOI:10.34229/2707-451X.21.3.6

E. MUSTAFAYEV, R. AZIMOV

COMPARATIVE ANALYSIS OF THE APPLICATION OF MULTILAYER AND CONVOLUTIONAL NEURAL NETWORKS FOR RECOGNITION OF HANDWRITTEN LETTERS OF THE AZERBAIJANI ALPHABET

Abstract. *The implementation of information technologies in various spheres of public life dictates the creation of efficient and productive systems for entering information into computer systems. In such systems it is important to build an effective recognition module. At the moment, the most effective method for solving this problem is the use of artificial multilayer neural and convolutional networks. This paper is devoted to a comparative analysis of the recognition results of handwritten characters of the Azerbaijani alphabet using neural and convolutional neural networks. The results of numerical experiments are given.*

Keywords: *neural networks, feature extraction, OCR.*

Introduction. Recognition of handwritten characters is an important problem of automation. At the moment, the most promising approach to solving this problem is the use of artificial neural and convolutional networks as classifiers. The main problem in the construction of such classifiers is the lack of a formal mechanisms for choosing the type and architecture of a neural network (NN). The efficiency of a classifier based on neural networks is determined by the architecture of the network, the number of layers, the nature of the layers and connections between them, the number of neurons in the layers. In the paper, a comparative analysis of the use of neural and convolutional neural networks [1 – 4] will be carried out on the example of the problem of recognizing handwritten characters of the modern Azerbaijani alphabet based on the Latin spelling [5 – 9].

Artificial multilayer feedforward neural networks.

A neural network is a network with a finite number of elements of the same type, analogs of neurons, with various types of connections between them. The basis of each neural network is made up of relatively simple, in most cases of the same type, elements that imitate the work of brain neurons.

Each neuron is characterized by its current state, by analogy with the nerve cells in the brain, which can be excited or inhibited. It has a group of synapses – unidirectional input connections connected to the outputs of other neurons, and also has an axon – the output connection of a given neuron, from which the signal (excitation or inhibition) is sent to the synapses of the following neurons.

Each input is multiplied by the corresponding weight, similar to synaptic strength, and all products are added up, determining the level of neuron activation.

Each weight corresponds to the "strength" of one biological synaptic connection.

The summing block corresponding to the body of a biological element determines the current state of the neuron as the weighted sum of its inputs (Fig. 1):

$$s = \sum_{i-1}^{n} x_i \cdot w_i + \theta,$$

where w_i – synapse weight $(i = 1..n)$, θ – bias value, s – summation result, x_i – input vector component (input signal) $(i = 1..n)$, n – number of input neurons.

The output of a neuron is a function of its state

$$y = f(s),$$

where y – output signal of the neuron, f – nonlinear transformation (activation function). The logistic function, hyperbolic tangent function, ReLU function or Leaky ReLU function can be taken as a nonlinear activation function [2–4].

In multilayer neural networks, neurons are combined into layers. A layer is a collection of neurons with a single set of input signals (Fig. 2). The number of neurons in each layer can be any and in no way connected in advance with the number of neurons in other layers. In general, the network consists of k layers, numbered from left to right. External input signals are fed to the inputs of the neurons of the first layer (the input layer is often numbered as zero), and the outputs of the network are the outputs of the last layer. The number of input and output elements is determined by the problem definition. In addition to the input and output layers in a multilayer neural network, there are one or more intermediate (hidden) layers. Multilayer networks can form in cascading layers. The output of one layer is the input for the next layer. The work of the neural network consists of transforming the input vector X into the output vector Y, and this transformation is done with the weights of the network.

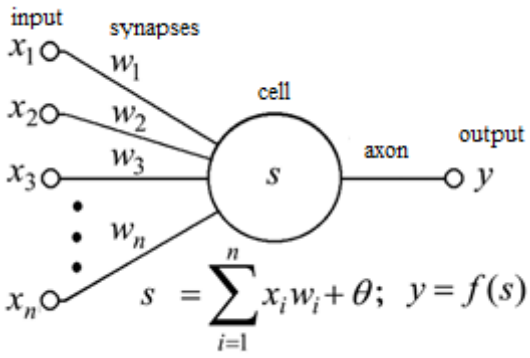


FIG. 1. Artificial neuron

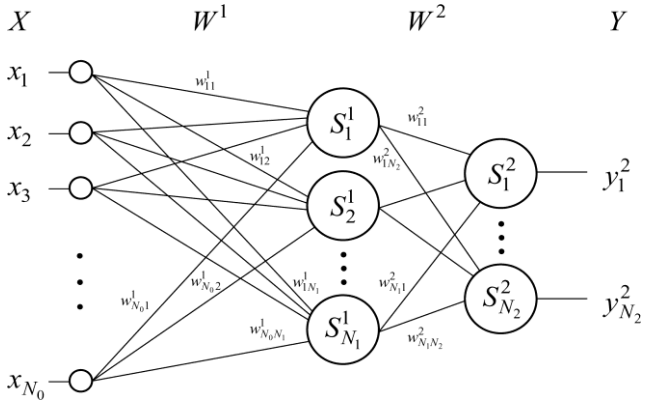


FIG. 2. Multilayer network

Convolutional neural networks. A convolutional neural network (CNN) is usually an alternation of convolution layers, subsampling layers, and in the presence of fully-connected layers at the output. In this work, we will use classical LeNet-5 architecture proposed by Yann Lecun [1] as a CNN (Fig. 3).

A convolutional layer is a collection of feature maps. Feature maps are two-dimensional matrices that represent the result of convolution by a separate filter. Each neuron of the feature map is connected to a part of the neurons of the previous layer. All maps of the convolutional layer are the same size and are calculated using the formula:

$$(w,h) = (W - k + 1, H - l + 1),$$

where (w,h) – feature map size, W and H – the width and height of the original image, k and l – width and height of convolution kernel.

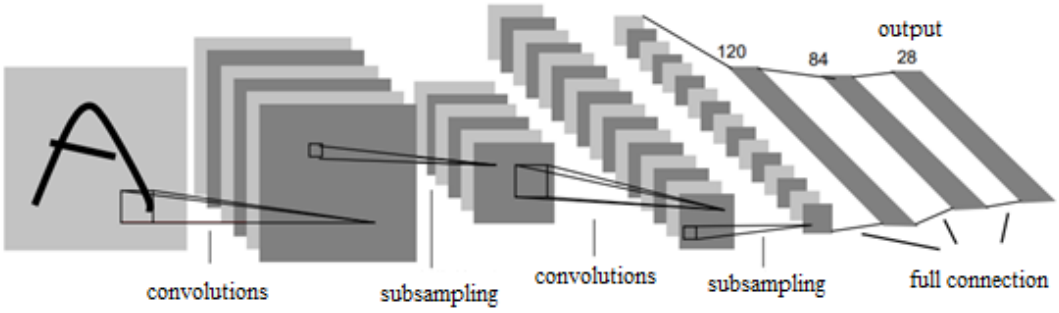


FIG. 3. Convolutional neural network

The convolutional layer is similar to the application of the convolution operation, where only a small size matrix of weights (the convolution kernel) is used, which is "transported" over the entire processed layer (Fig. 4).

The kernel is a system of shared weights. Within one feature map, all neurons use the same weights. In the convolutional layer, the total weights reduce the number of connections and allow finding the same feature over the entire image area.

Kernel size $k \times l$ traverses with a given step (usually 1) the entire image, at each step, element-by-element multiplies the contents of the window by the kernel matrix, the result is summed up and written into the result matrix (Fig. 4). Then the result matrix is passed through the activation function (usually ReLU) and this forms the output of the convolutional layer.

The subsumpling layer, like the convolutional layer, has maps and their number coincides with the previous (convolutional) layer. The purpose of a layer is to reduce the dimensions of the maps of the previous layer. Basically, there are two types of subsampling layer: selection of maximum (max pooling) or average (average pooling) (Fig. 5).

For this, the feature map of the previous layer is divided into cells of a certain size (usually 2 x 2). Further, for each cell, depending on the selected downsampling algorithm, the maximum or average cell value is selected.

After sequential alternation of the convolutional and subsampling layers, the output of the last pooling layer is fed to the input of a fully connected 3-layer feedforward neural network to directly implement the classification function. The number of neurons in the output layer is determined by the nature of the task and is usually equal to the number of recognized classes.

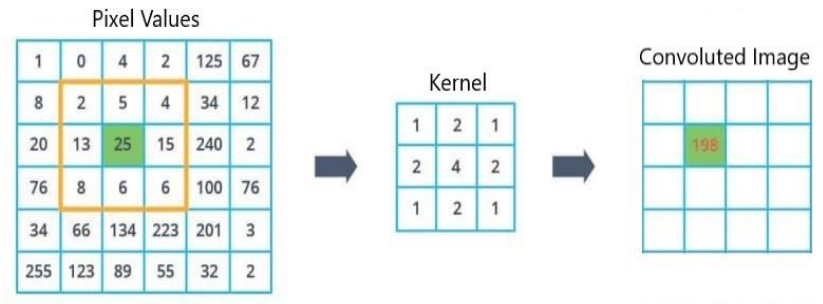


FIG. 4. Convolutional map values

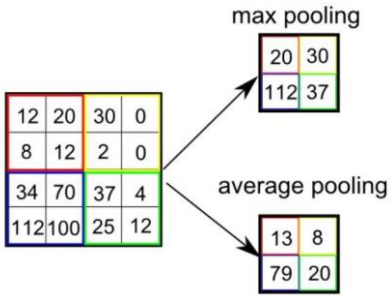


FIG. 5. Pooling layer

Conditions for carrying out numerical experiments

For comparative analysis handwritten characters of the Azerbaijani alphabet, consisting of 32 characters, were used as the object of recognition (Fig. 6):

{ABCÇDEƏFGĞHIİJKLMNOÖPQRSŞTUÜVXYZ}.

From this set of characters symbols with strokes and dots at the top are removed – {ĞİÖÜ}, since their recognition falls into two stages: recognition of the upper and lower parts. The lower part is determined using the character recognition module {GIOU}, and the upper part can be done using some other "lightweight" algorithms. So, the number of classes for recognition will be 28.

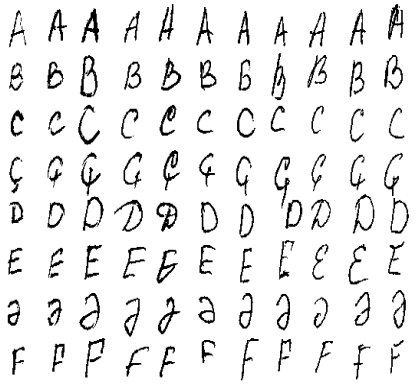


FIG. 6. Example of handwritten characters

For training and testing we used a database of handwritten characters scaled up to size 20×20 with preserve original proportions. When training convolutional networks, we used a 32×32 size with a 20×20 character in the center.The database for each class contains 500 copies for training and 100 copies for testing, which ultimately gave $28\times500=14000$ copies in the training sample and $28\times100=2800$ copies in the test sample. To expand the training database, the augmentation technique was used – the artificial generation of new symbols using affine transformations. Using this technique, character bases consisting of 28000, 42000, and 70000 characters were generated to conduct numerical experiments as a test sample.

For completeness of the experiment, when training multilayer network, we fed not only the image itself (a vector with a dimension of 400) to the network input, but also the results of feature extraction. The Peripheral Directional Contributivity (PDC) is used as feature extraction method (Fig. 7) [10]. This feature reflects well the complexity, the orientation and the relative positioning of strokes in symbols. Directional Contributivity (DC) of each point of a symbol represents 8 (or 4)-dimensional vector. Each component of a vector represents distance from this point up to a symbol border in one of possible 8 directions (or a maximum of distances on 4 directions: vertical, horizontal and two diagonal). Then values of a vector are normalized. At movement from border on one of four directions we shall meet a point in which white color passes in black. We shall name such point as the 1st order peripheral point (or the peripheral point of depth 1). If we move further, we shall meet the 2nd order peripheral point.

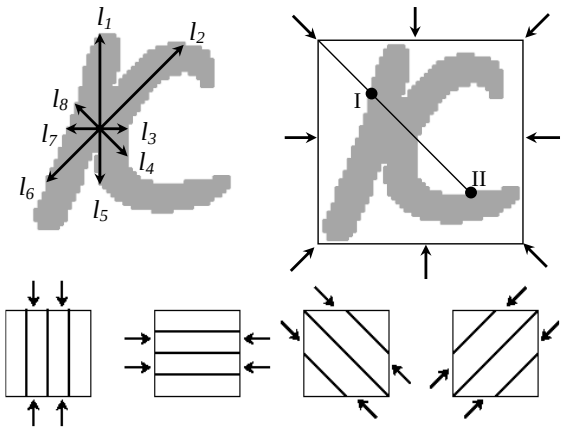


FIG. 7. Extraction of PDC feature

As a feature, we use DC for all the 1st and 2nd order peripheral points on four directions. For reduction of dimension and sensitivity shifts, we divide the sequence of peripheral points in each direction into segments and use the average DC values in each segment. In our experiments, we used the following PDC parameters:

- Dimension DC = 4;
- Directions quantity = 4 (two each in the horizontal and vertical directions);
- Depth quantity = 2;
- Number of segments = 8.

As a result, the dimension of the PDC feature will be $4\times4\times2\times8=256$.

For a comparative analysis, the following NN models were used.

- *Artificial multilayer feedforward neural networks (AMFNN).*
  When training models 1 and 2, an image of a symbol without feature extraction was fed to the input of the neural network, and for models 3 and 4 – the results of feature extraction. Also, these models differ in the number of hidden layers.

N	Model	Number of layers	Number of neurons on layers	Number of parameters
1	Pixels_400_50_28	3	400–50–28	21478
2	Pixels_400_50_50_28	4	400–50–50–28	24028
3	PDC_256_50_28	3	256–50–28	14278
4	PDC_256_50_50_28	4	256–50–50–28	16828

- *Convolutional neural networks (CNN).*
  These models differ in the number of feature maps in the "second" layer.

N	Model	Number of feature maps on the 1st layer	Number of feature maps on the 2nd layer	Number of neurons on fully connected layers	Number of parameters
1	LeNet5-filters-6-16	6	16	120–84–28	63236
2	LeNet5-filters-6-8	6	8	120–84–28	38028
3	LeNet5-filters-6-4	6	4	120–84–28	25424

For the software implementation of neural networks, the Keras library based on the Tensorflow framework was used [11]. To minimize the objective function, we used the Adam algorithm with standard parameters [12 – 13].

Each neural network was trained 5 times using different starting points and the result was taken the variant that gave the maximum result on the test sample.

The comparison of recognition results for a test sample of neural network models was carried out according to the following parameters:

- the volume of the training sample (14000, 28000, 42000 и 70000 symbols),
- selection of a method in a pooling layer (maximum or average),
- influence of feature extraction,
- influence of the number of feature maps in the "second" layer.

As a result, for comparative analysis, we will deal with 40 neural networks:

- AMFNN: {quantity of models}x{quantity of DB} = 4 x 4 = 16;
- CNN: {quantity of models}x{quantity of DB} x {quantity of pooling methods} = 3 x 4 x 2 = 24.

Results of numerical experiments. Below are the results of the experiments.

TABLE. 1. Recognition results for all neural networks

N	Model	Training data	Number of parameters	Recognition on training data	Recognition on test data
1	LeNet5-filters-6-16	14000	63236	99,64 %	89,32 %
2	LeNet5-filters-6-16	70000	63236	99,58 %	89,14 %
3	LeNet5-filters-6-16	42000	63236	99,23 %	88,82 %
4	LeNet5-filters-6-8	70000	38028	99,26 %	88,75 %
5	LeNet5-filters-6-8	14000	38028	99,57 %	88,57 %
6	LeNet5-filters-6-8	28000	38028	99,38 %	88,43 %
7	PDC-256-50-28	14000	14278	99,50 %	88,21 %
8	LeNet5-filters-6-4	70000	25424	99,02 %	88,18 %
9	LeNet5-filters-6-16	28000	63236	99,49 %	88,18 %
10	LeNet5-filters-6-4	28000	25424	99,21 %	88,07 %
11	LeNet5-filters-6-4	42000	25424	99,22 %	87,96 %
12	LeNet5-filters-6-8	42000	38028	99,53 %	87,96 %
13	PDC-256-50-28	70000	14278	97,79 %	87,86 %
14	PDC-256-50-50-28	42000	16828	98,58 %	87,54 %
…	…	…	…	…	…
27	Pixels-400-50-28	14000	21478	99,94 %	82,04 %
28	Pixels-400-50-28	28000	21478	99,49 %	81,46 %

** For a compact view of this table for convolutional networks, the results of models with different downsampling methods are not shown, but the maximum values for each model are selected.*

TABLE. 2. Recognition results for multilayer neural networks

N	Model	Training data	Number of parameters	Recognition on training data	Recognition on test data
1	PDC-256-50-28	14000	14278	99,50 %	88,21 %
2	PDC-256-50-28	70000	14278	97,79 %	87,86 %
3	PDC-256-50-50-28	42000	16828	98,58 %	87,54 %
4	PDC-256-50-50-28	70000	16828	97,97 %	87,50 %
5	PDC-256-50-28	42000	14278	98,69 %	87,32 %
6	PDC-256-50-50-28	14000	16828	99,60 %	87,32 %
7	PDC-256-50-28	28000	14278	98,53 %	87,29 %
8	PDC-256-50-50-28	28000	16828	98,74 %	87,29 %
9	Pixels-400-50-50-28	70000	24028	98,30 %	84,04 %
10	Pixels-400-50-28	70000	21478	98,76 %	83,96 %
11	Pixels-400-50-50-28	42000	24028	98,91 %	83,36 %
12	Pixels-400-50-50-28	14000	24028	99,98 %	83,25 %
13	Pixels-400-50-28	42000	21478	99,20 %	83,04 %
14	Pixels-400-50-50-28	28000	24028	99,27 %	82,43 %
15	Pixels-400-50-28	14000	21478	99,94 %	82,04 %
16	Pixels-400-50-28	28000	21478	99,49 %	81,46 %

TABLE. 3. Comparative results according to subsampling methods for CNN

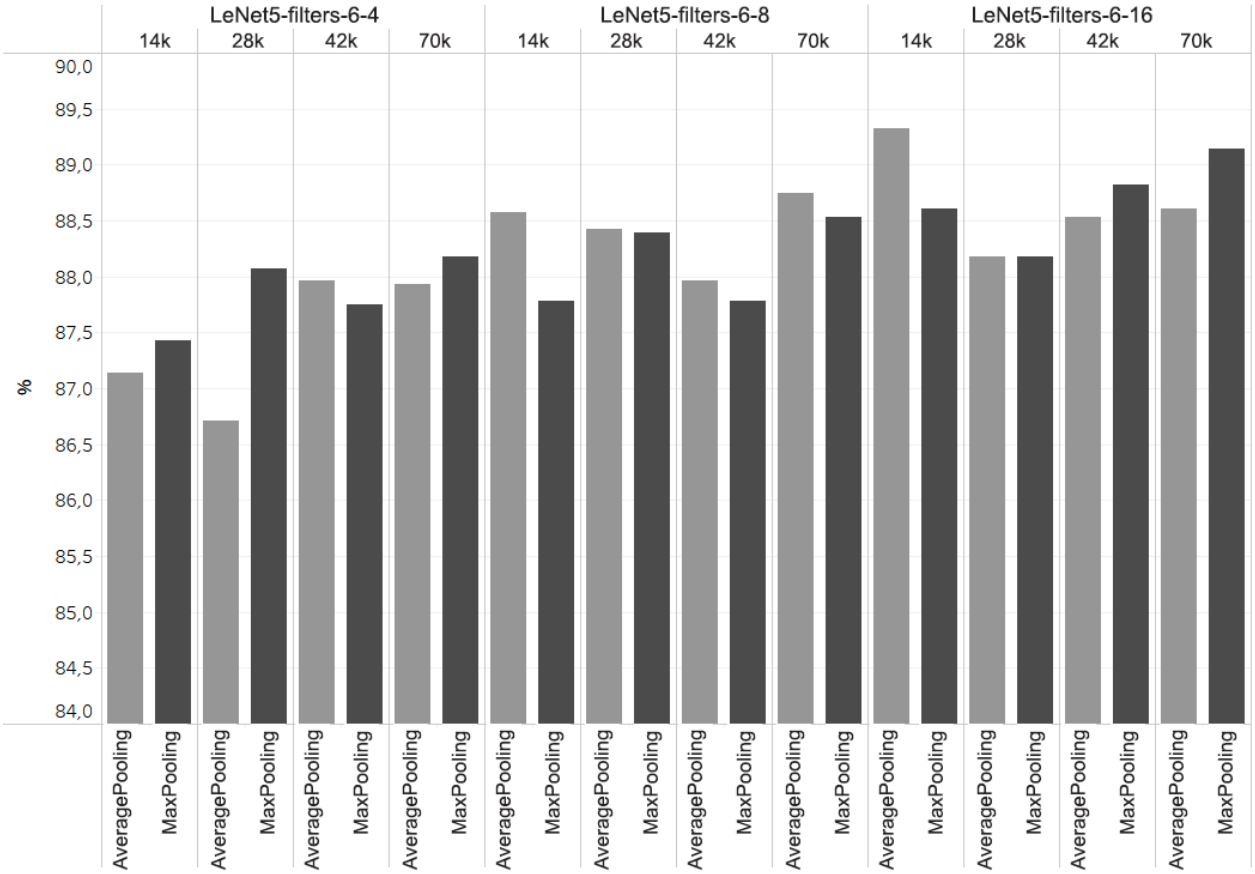
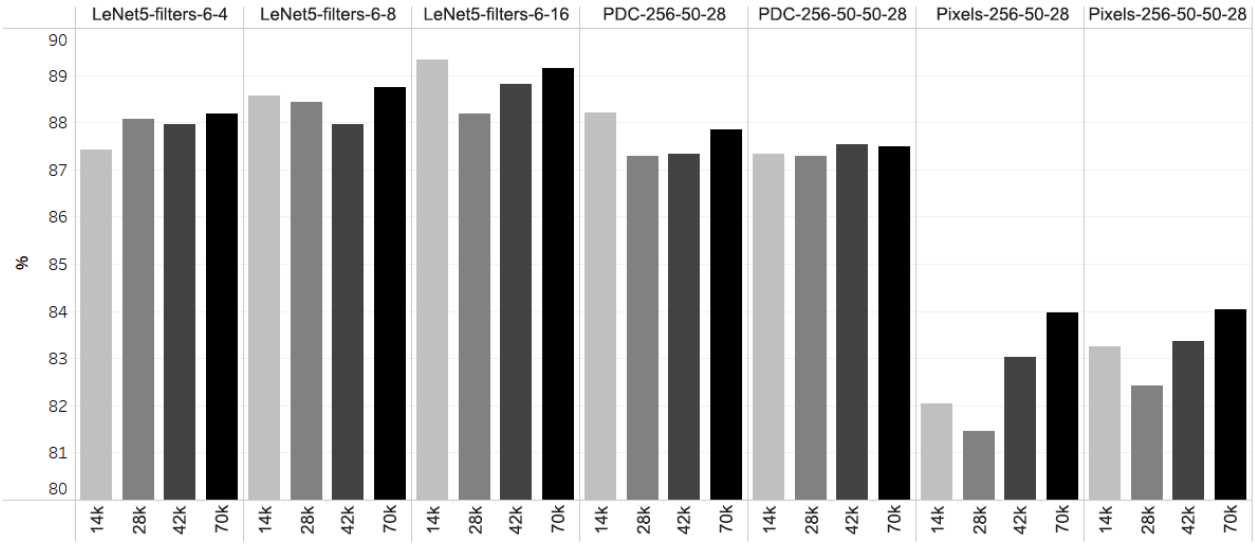


TABLE. 4. Comparative results depending on the volume of the training database



Conclusion. As expected, the classical convolutional network with 6–16 feature maps showed better results, but the multilayer network, which trained on the symbols images, had the worst performance. Note that a multilayer network trained on the results of feature extraction showed rather high results, comparable with convolutional networks. In general, convolutional networks performed better than multilayer networks (Table. 1).

As expected, neural networks, which were trained on the results of feature extraction, showed higher results than networks that were trained on the image itself (Table. 2).

As can be seen from Table. 3, there is no definite advantage in the choice of the method in the subsampling layer. The choice of the subsampling method for a particular model can be selected experimentally.

Increase the training database did not give a tangible improvement in recognition results for convolutional networks and networks with preliminary feature extraction. However, for networks learning without feature extraction, an increase in the size of the database led to a noticeable improvement in performance.

References

1. Lecun Y., Bottou L., Bengio Y. Haffner P. Gradient-based learning applied to document recognition. *Proc. of the IEEE*. 1998. **86** (11). P. 2278–2324. https://doi.org/10.1109/5.726791
2. Golovko V.A., Krasnoproshin V.V. Neural network data processing technologies. Minsk: BSU, 2017. 263 p. (in Russian)
3. Osovsky S. Neural Networks for Information Processing. *Finance and statistics*. 2002. 344 p. (in Russian)
4. Mustafayev E.E. Handwritten text recognition methods. Fuyuzat, 2020. 189 p. (in Russian)
5. Aida-zade K.R., Mustafayev E.E. Intelligent recognition system of Azerbaijani handwritten forms. Proc. The Scientific Conference *"Modern problems of Cybernetics and Information Technologies"*. Vol. III. Baku. 2006. P. 85–88. (in Russian)
6. Aida-zade K.R., Mustafayev E.E. About one hierarchical handwritten recognition system on the bases neural networks. *Transactions of the NAS of Azerbaijan, series of PTMS*. 2–3. 2002. P. 94–98. (in Russian)
7. Aida-zade K.R., Mustafayev E.E., Hasanov J.Z. About knowledge base usage for increasing intellectuality of recognition systems, Proc. the 11th Russian Conference *"Mathematical Methods of Pattern Recognition"*. 2003. Moscow. P. 6–8. (in Russian)
8. Mustafayev E.E. Hierarchical Multilevel Form Recognition System. Proceedings of scientific conference *"Modern problems of applied mathematics"*. Baku. 2002. P. 154–157.
9. Aida-zade K.R., Mustafayev E.E. Intelligent handwritten form recognition system based on artificial neural networks. Proceedings of the Intern. Conf. on Modeling and Simulation, 2006, 28-30 August, Konya, Turkey. P. 609–613.
10. Arif A.F., Takahashi H., Iwata A., Tsutsumida T. Handwritten postal code recognition by neural network – a comparative study. *IEICE Trans.Inf.&Syst*. 1996. **E79-D** (5). P. 443–449.
11. Francois Ch. Deep Learning with Python. Manning Publications, Shelter Island, NY. 362 p.
12. Eldan R., Shamir O. The power of depth for feedforward neural networks. Conference on Learning Theory. 2016. **49**. P. 907–940.
13. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234 (accessed: 26.07.2021)

Received 28.07.2021

Elshan Mustafayev,
PhD, leading researcher, Institute of Control Systems of the NAS of Azerbaijan, Baku,
https://orcid.org/0000-0002-1544-3897
elshan.mustafayev@gmail.com

Rustam Azimov,
researcher, Institute of Control Systems of the NAS of Azerbaijan, Baku.
https://orcid.org/0000-0002-4554-6985
rustemazimov1999@gmail.com

УДК 004.852

Е. Мустафаєв [*], Р. Азімов

Порівняльний аналіз застосування багатошарових і згорткових нейронних мереж для розпізнавання рукодрукованих літер на прикладі азербайджанського алфавіту

Інститут Систем Управління НАН Азербайджану, Баку
[*] Листування: elshan.mustafayev@gmail.com

Вступ. Впровадження інформаційних технологій у різних сферах суспільного життя диктує створення ефективних і продуктивних систем введення інформації в комп'ютерні системи. В таких системах важливе значення має побудова ефективного розпізнавального модуля. На даний момент найбільш перспективним підходом до вирішення цього завдання є використання штучних багатошарових і згорткових нейронних мереж.

Мета роботи. Провести порівняльний аналіз результатів розпізнавання рукодрукованих символів азербайджанського алфавіту за допомогою багатошарових і згорткових нейронних мереж.

Результати. Проведено аналіз залежності результатів розпізнавання від наступних параметрів: архітектури нейронних мереж, розміру навчальної бази, вибору алгоритму субдискретизації, використання алгоритму виділення ознак. Для збільшення навчальної вибірки використана техніка аугментації зображень. На основі реальної бази з 14000 символів були утворені бази по 28000, 42000 і 72000 символів. Наведено опис алгоритму виділення ознак.

Висновки. Аналіз результатів розпізнавання на тестовій вибірці показав:

- як і очікувалося, згорткові нейронні мережі показали більш високі результати, ніж багатошарові нейронні мережі;
- класична згорткова мережа LeNet-5 показала найбільш високі результати серед всіх типів нейронних мереж. Однак, багатошарова 3-х шарова мережа, на вхід якої подавали результати виділення ознак, показала досить високі результати, які можна порівняти зі згортковими мережами;
- немає певної переваги у виборі методу в субдискретному шарі, вибір методу субдискретизації (max-pooling або average-pooling) для кожної моделі може бути підібраний експериментальним шляхом;
- збільшення навчальної бази даних для даної задачі не дало відчутного поліпшення результатів розпізнавання для згорткових мереж і мереж з попереднім виділенням ознак. Однак для мереж, що навчаються без виділення ознак, збільшення розміру БД призводило до помітного поліпшення показників.

Ключові слова: нейронні мережі, виділення ознак, розпізнавання символів.