Electronic Letters on Computer Vision and Image Analysis 21(1):102-110; 2022

A neural network with competitive layers for character recognition

Alexander Goltsev, Vladimir Gritsenko

International Research and Training Center of Informational Technologies and Systems,
National Academy of Sciences of Ukraine, Kiev, Ukraine

Received 5th of February, 2021; accepted 26th of May 2022

Abstract

A structure and functioning mechanisms of a neural network with competitive layers are described. The network is intended to solve a classification task with a limited number of classes. The network consists of several competitive layers of neurons. Each layer is a neural network consisting of a number of neurons represented as a layer. The number of neural layers is equal to the number of recognized classes. All neural layers have one-to-one correspondence with one another and with the input raster. The neurons of every layer have mutual lateral learning connections, which weights are modified during a learning process. There is a competitive (inhibitory) relationship between all neural layers. This competitive interaction is realized by means of a "winner-take-all" (WTA) procedure which aim is to select the layer with the highest level of neural activity.

Validation of the network has been done in experiments on recognition of handwritten digits of the MNIST database. The experiments have demonstrated that its error rate is few less than 2%, which is not a high result, but it is compensated by a very simple structure and functioning mechanisms and rather fast data processing.

Key Words: Pattern Recognition, Neural network, Character recognition, Learning.

1 Introduction

The article proposes functional mechanisms of a neural network with architecture of a competitive layer models (CLM). The CLM was first introduced as a model for spatial feature linking in [1]. Independently, the same CLM architecture was described in [2], [3] under the title "Recurrent neural network". According to its name, the competitive layer model consists of several layers of neurons. Each such neural layer is a separate neural network represented as a layer.

In general, the competitive layer models are proposed for grouping and clustering data of different types. A more concrete implementation of such grouping and clustering tasks, performed by the competitive layer models, are the following: feature linking, sensory and figure-ground segmentation [1], [4]; texture discrimination and image segmentation [5], [6]; detecting brain activated regions from fMRI data [7]. In [8], a rigorous analysis of the convergence of CLM is presented, in which it is proved that the CLM with one winner in each row and each column converges. In [2], [3], the neural networks with the CLM-like architecture were used to separate line drawings (handwritten characters) into strokes of different orientations. The same CLM-like architecture is used in [9], where the network solves the task of improving the shapes of texture segments extracted by a universal algorithm which is developed to segment any input image into a number of homogeneous fine-grained texture segments.

In this article, the working capacity of the proposed neural network is tested on the task of recognition of handwritten characters. The problem of recognizing handwritten characters arose in the middle of the last

Correspondence to: <root@adg.kiev.ua>

Recommended for acceptance by Dr. Angel D. Sappa
https://doi.org/10.5565/rev/elcvia.1392
ELCVIA ISSN: 1577-5097
Published by Computer Vision Center / Universitat Autonoma de Barcelona, Barcelona, Spain

century and continues to be still actual. A review on work done for the character recognition before 1990 is presented in [10]. The Hindi character recognition algorithm based on the Devnagari script is described in [11]. Handwritten character recognition of the MNIST database has been studied and discussed in [12]. There is a more recent overview of handwritten character recognition of the MNIST database in [13]. The application of the deep learning neural network for the same task is described in [14]. In [15], a neural network model is proposed for visual recognition (in particular, for character recognition). This work is very close to ours both in basic ideas and in using the same database (MNIST), and even in the same network performance.

In the field of machine learning and character recognition, the MNIST database is widely used to test and evaluate various recognition algorithms and systems [16]. Therefore, we use the MNIST database to test and evaluate the performance of the proposed neural network.

As mentioned above, the competitive layer neural network consists of several layers of neurons. All neural layers are in competitive interrelations between each other. Each neural layer serves to represent a certain class, group, cluster, image segment, and so on. Every neuron of each neural layer interacts with other neurons of the same layer by means of excitatory and inhibitory connections. In the classical CLM, each neuron of every layer laterally interacts with all other neurons of the same layer by means of a complex function which is excitatory for short distances and weakly inhibitory for larger distances. The competitive layer neural network works using an iterative procedure that involves the interaction of neurons within each layer through lateral connections and competitive relationships between the layers. The competition between the network layers is realized in operating neural columns, each of which crosses all neural layers and includes all the corresponding neurons of different layers. In each operating neural column, the "winner-take-all" procedure (WTA) selects the neuron with the highest activity. In each operating neural column, the selection is performed independently of all others.

In this work, the neural network with competitive layers is applied for solving the character recognition tusk, uses a learning procedure to form a system of lateral connections between neurons of the same layers. There is no learning between the neurons of different layers.

In our previous works, we applied modular assembly neural networks for solving the same character recognition problem (e.g., [17], [18], [19]). It is worth noting that the competitive layer networks can also be categorized as modular neural networks. Our previous modular neural networks used specially designed features to describe and recognize characters, so that, any image processing started by extracting these features from the input image. The most effective of these features are the LiRA-features (e.g., [20], [21], [22]). Unlike that, the proposed competitive layer network does not use any pre-formed features, but uses raw image as an input and creates adaptive features during the learning procedure within connection structures of the layers. This is due to the specific architecture and functional mechanisms of the layered network that demand using only local features associated with certain places of the recognized object.

The article is organized as follows. Section 2 presents a general description of the neural network functioning. Section 3 describes a learning procedure that forms a system of lateral connections between neurons of all layers. Section 4 demonstrates experimental validation of the neural network with competitive layers performed on handwritten digits of the MNIST database. Section 5 is devoted to discussion and conclusions.

2 A general description of the neural network functioning

In the present work, the neural network with competitive layers consists of several neural layers which number equals to the number of classes to be recognized; let us designate this number by K.

It is assumed that all images of characters are represented on a raster of $N = I \times J$ of binary pixels. Let us introduce a binary matrix $\mathbf{R}[i][j]$ of $N = I \times J$ size to represent an input image: the pixels belonging to the input character are represented by one-valued elements of the matrix $\mathbf{R}$, other pixels are zero-valued. The same binary input image $\mathbf{R}$ is fed to all layers of the network.

Every neural layer is intended to represent all pixels of a certain character class and contains such number of neurons that is equal to the number of image pixels: $N = I \times J$. So that, each neuron of every neural layer corresponds to one pixel of the input image. In other words, all neural layers have one-to-one correspondence with the input raster and with one another. Figure 1 illustrates this description.

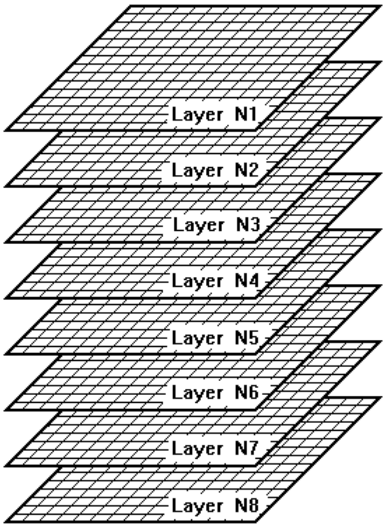

Figure 1. A schematic picture of a neural network with competitive layers. In the figure, the network is depicted as consisted of 8 neural layers.

Each neuron of the network has a linear output characteristic, that is, any excitation at the input of the neuron is immediately transformed into the same output activity. Let us introduce an integer matrix $\mathbf{E}(k)[i][j]$ of ($N = I \times J$ size) to represent gradual (integer) values of output neural activity of all neurons of the k-th layer.

There are connections with variable weights between all neurons of the same layer in each layer of the network. These weights change during the learning process. Let us introduce an integer matrix $\mathbf{W}(k)[i][j]$ to represent all connection weights between all neurons of the k-th layer. In the formulas presented below, the connection weight $w(k)[i][j]$ denotes the weight of connection that is directed from the i-th neuron of the k-th layer to the j-th one of the same layer.

At the beginning of the recognition procedure, the same input image represented in the matrix **R** is fed to all competitive layers of the network simultaneously. As a result of that, this input excitation is transformed into initial neural activity in all K neural layers:

$$\mathbf{E}(k)[i][j] = \mathbf{R}[i][j], \quad (1)$$

where $i = 0, 1, 2, ..., I$; $j = 0, 1, 2, ..., J$; $k = 0, 1, 2, ..., K$.

After that, a one-step calculation procedure is performed inside each layer. The procedure consists of interactions between all neurons of the same layer through learned connections. More exactly, the neurons having initial neural activity represented in the matrix **E** spread their influence in all K layers of the network through learned connections that have different weights in all layers. Consequently, a secondary neural activity appears in the network - new values of neural activity **E** are calculated as follows

$$\mathbf{E}(k)[i][j] = \sum_{i=0}^{I} \sum_{j=0}^{J} \mathbf{E}(k)[i][j]\, w[i][j]. \quad (2)$$

Spreading initial neural activity within the network layers by means of lateral learned connections exacts has a strong effect on the levels of neural activity at all network layers.

There is a competitive (inhibitory) relationship between all neural layers. This competitive interaction is realized by means of a "winner-take-all" (WTA) procedure which aim is to select the layer with the highest level of the secondary neural activity. To perform the WTA concurrent interaction, all the secondary neural activity of each layer is first summarized. Let us introduce a vector of summarized neural activity of all K layers of the network by $\mathbf{H}[k]$; it is calculated by the formula:

$$\mathbf{H}(k) = \sum_{i=0}^{I} \sum_{j=0}^{J} \mathbf{E}(k)[i][j]. \tag{3}$$

The layer with the highest level of secondary neural activity $H^{\max}$ is selected as follows

$$H^{\max} = \underset{k=0}{\overset{K}{\mathbf{MAX}}}\ H(k). \tag{4}$$

And the number of this highest activity layer Z which is a label or a number of the character class is determined by the formula

$$Z = \arg H^{\max} \tag{5}$$

Here, it is worth to note, that as distinct from the competitive layer network presented in [2] and [3], where inhibitory relationship exists between the corresponding neurons in each operating neural column, in the present neural network competitive interaction is based on integral neural activity in different neural layers. The reason is that the experiments have shown the advantage of the current WTA procedure over the older one (in this task).

Also, the recognition procedure in the present network is performed as a result of a single-step computation of neural activity compared to using an iterative multi-step procedure in the networks described in [2] and [3]. This obviously increases the speed of the computation process.

3 A learning procedure

The weights of connections inside the network layers are modified according to the Hebb's learning rule [23].

In the learning stage, all images of the available training set are fed to the layer network in turn. Let us consider a learning process of one such input image. Let the input image belong to some class C. The layer network recognizes this image as described above. If the input image is correctly recognized, i.e., if $Z = C$, then no modification of any connection weights is performed.

However, if it turns out that as a result of the recognition procedure some other D-th layer has the highest level of neural activity ($Z = D$), i.e., the network recognizes the input image as belonging to some wrong class D ($D \neq C$), then the weights of connections are changed in two layers of the network C and D as follows

$$\mathbf{W}(C)[i][j] = \mathbf{W}(C)[i][j] + \Delta W, \tag{6}$$

$$\mathbf{W}(D)[i][j] = \mathbf{W}(D)[i][j] - \Delta W, \tag{7}$$

where ΔW is a connection weight increment value.

Then the next training sample is fed to the network, it is recognized according to above procedure and necessary connection modifications are performed.

In the learning stage, all training samples are processed by the network many times in turn, one complete cycle after another. In the field of learning algorithms, a complete cycle of successive consideration of all samples of the training set is named as epoch. Sequential consideration of all training samples continues many times, epoch by epoch, until the convergence of the network is achieved, that is, until all samples of the training set are classified correctly.

It follows from Eqs. (6), (7) that the connection weights of the matrix $\mathbf{W}$ can have both positive and negative values.

In [20], [21], [22], a defense space mechanism was introduced, which was used in [17], [18], [19]. The usage of this mechanism usually results in improved generalization capabilities and in an increase of the percentage of correct classification. Therefore, in this work, we also use a defense space mechanism according to the following description.

As described above, the procedure for recognition of some current input training sample leads to formation of the vector of summarized neural activity in all K layers of the network – $\mathbf{H}[k]$. Let the input image belong to the class C. In the vector $\mathbf{H}[k]$, the value of its C-th component $\mathbf{H}[C]$ is artificially reduced; let us denote this new reduced value as $\mathbf{H}^*[C]$: $\mathbf{H}^*[C] = \mathbf{H}[C]\ (1 - T)$, where T is the "defense parameter", $0 \le T < 1$. Then, the reduced level of the layer activity $\mathbf{H}^*[C]$ is compared with the activity levels of all other layers to determine the highest one according to Eq. (4). The number (class) of the highest layer-winner is calculated by Eq. 5. Then the same procedure for modifying the connection weights is used according to Eqs. (6)–(7).

4 Experiments

As mentioned above, validation of the neural network with competitive layers is performed on handwritten digits of the MNIST database. MNIST is a database of labeled handwritten digits, with separate training and test sets, and therefore is an easily interpretable domain that allows a fast comparison between different techniques. There are 60 000 images of digits in the training set of the MNIST database and 10 000 images in its test set. The size of each image is 28×28 pixels. Each pixel of the image can take an integer brightness value varying from 0 to 255.

In the first step of preprocessing, every MNIST image, consisting of integer brightness pixels, is converted into binary one by means of a simple thresholding operation. In the second step, binary contours of the digits are detected in this binary image. The binary contours of the digit are represented as one-valud pixels of the binary matrix $\mathbf{R}$ (introduced in Section 2). These contours serve as inputs to the network. Figure 2 illustrates the preprocessing. The top line demonstates the binary images of five training samples of the MNIST database after the thresholding operation; the bottom line shows the detected contours of these digits.

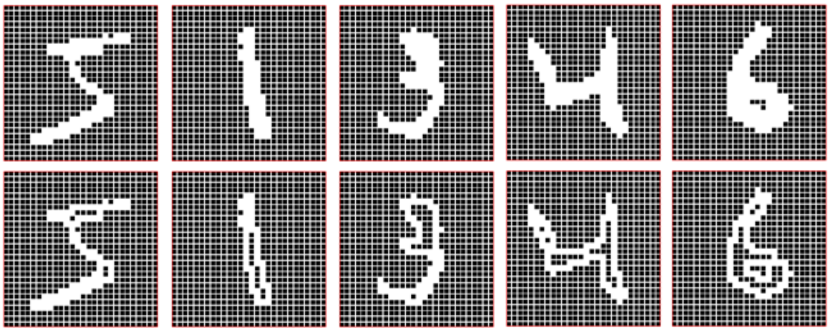

Figure 2. Two steps of preprocessing of every image of the MNIST database (from top to bottom).

Many classifiers listed on the MNIST website [16], that achieve high recognition rates use distortions of the original shapes of the training characters. Due to such distortions, the number of available training samples may be considerably increased what usually leads to an increase in the percentage of recognition. We also applied 10 simple distortions of each training sample for the learning process. These distortions are: small shifts (one pixel) of characters up, down, left and right; small slopes of characters to the left and right.

At first, the network was trained in turn on all various distortions of the original digits of the MNIST training set. After convergence of the network at each current distortion of all samples of the training set, the network recognition ability was measured using the test set. The convergence at each distortion usually took several tens of training epochs. Let us underline that for each convergence, we required zero errors on all 60 000 samples of the training set. The network connection weights (matrix $\mathbf{W}$) were not reset to zero after every convergence, but remained with the trained values. In Fig. 3 a typical experimental graphic shows the dependence of the network error rate on the number of distortions.

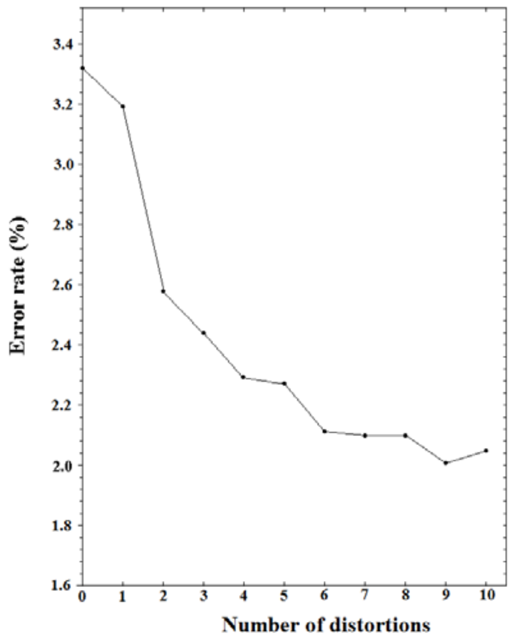

Figure 3. Dependence of the network error rate on the number of distortions.

As seen in Fig. 3, the error rate decreases with increasing the number of distortions, but not linearly. The best result in the graphic is achieved after learning the network for 9 distortions. It is not expected that the number of the network errors somewhat increases after learning on one more distortion.

Then we begun to study the dependence of the error rate on the value of the parameter T, using all 10 distortions of the entire training set of the MNIST database for learning. Namely, we gradually increased the value of the parameter T, and after convergence of the learning procedure at the current value of the parameter T, the network recognition ability was measured using the test set. Figure 4 demonstrates dependence of the network error rate upon the values of parameter T. The best result in the graphic is 187 errors, it is achieved at $T = 0.03$ (3%).

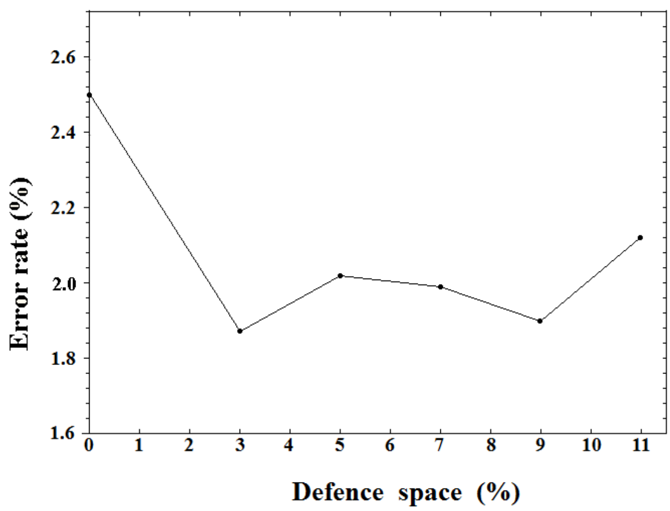

Figure 4. Dependence of the network error rate upon the values of parameter T.

It is evident from this figure, that a further increase in the value of the parameter T (more than T = 0.11 (11%) does not make sense for reducing the network error rate.

The best recognition rate achieved in the experiments is 187 errors for 10 000 examples of the MNIST test set. So, the performance of the present neural network with competitive layers is comparable to the performance of standard feed-forward networks, trained end-to-end with a back-propagation algorithm [16].

5 Discussion and conclusion

In this article, the neural network with competitive layers is presented. This network is actually a neural network classifier which is intended to recognize not a large number of classes. Indeed, since each recognized class is represented by a corresponding neural layer in the network, an increase in the number of recognized classes obviously leads to a proportional increase in the computational costs of data processing.

An experimental validation of the network is accomplished on the MNIST database. Testing has shown that the network error rate is few less than 2%. This result is not high in comparison with the top performances presented in the tables of [16] and [14]. As seen in these tables, the top recognition results demonstrate the deep learning networks trained on a distorted MNIST database.

As mentioned in [15], the success of deep neural networks trained with back-propagation leads to the widespread opinion that neurobiology-inspired plasticity rules are less efficient than the back-propagation algorithm. From the other hand, it is generally accepted that the back-propagation is biologically implausible. In the present work we use the learning rule, which is of biological plausibility, and which corresponds to Hebb’s idea that change of the synapse strength should be local – i.e. should depend only on the activities of the pre and post synaptic neurons.

Also, the proposed neural network has a significant similarity in architecture with biological neural networks, namely with the primary visual cortex of the mammalian brain, which is also a certain construction of neural layers and neural columns [24], [25]. And therefore, the proposed network might pretend to be biologically plausible.

In [12], it is mentioned that the human error rate is about 1.5% in the NIST database, which is one of two sources of the MNIST database. The human recognition rate is so low because there are many images in the NIST and MNIST databases (both in training and test sets) that cannot be unambiguously classified. Figure 5 demonstrates examples of such images. The number below each image indicates the class number to which the image belongs, as implied in the database.

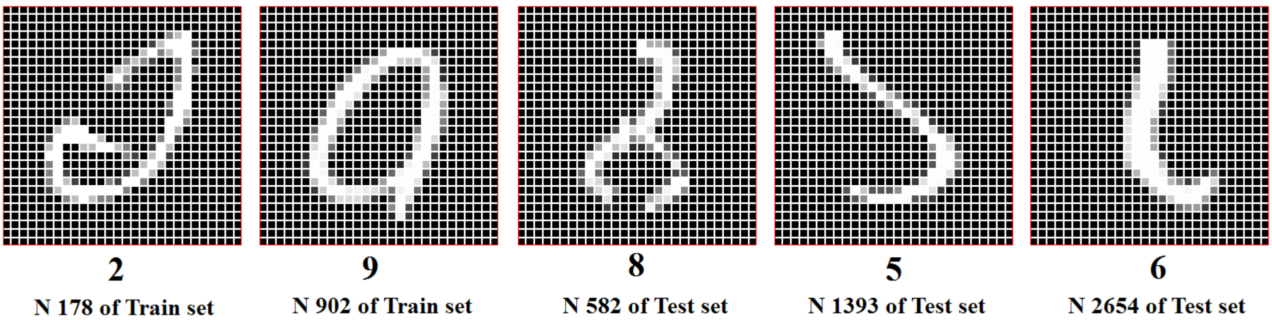

Figure 5. Some examples of ambiguous characters of the MNIST database.

The fact is that a human recognition rate on the MNIST database is rather low in comparison with many modern recognition devices. For example, in [14] an algorithm is described that demonstrates only 0.18% test error rate. In general, the recognition rate achieved by a classifier depends, first of all, on the feature set used for description of the objects to be recognized. The better the feature set, the higher the performance of the classifier. Moreover, in the field of pattern recognition it is well known that influence of the feature set on the recognition rate is much stronger than the type of classifier. Obviously, the top classifiers that are able successfully recognize ambiguous characters of the MNIST database do that by means of features that are radically different from those that humans use for character recognition.

In our previous works, the modular neural networks were applied for recognition of characters of the same MNIST database (e.g., [17], [18], [19]). The networks used the LiRA-features (e.g., [20], [21], [22]) for description of the character classes. The LiRA-features were not formed in the learning process; they were generated artificially and preliminarily. Usage of the LiRA-features led to good recognition results on the MNIST database. The LiRA-features have such an advantage as direct using of feature coordinates. However, this advantage is available only for centered images with normalized size of objects, such as digit images of the MNIST database. For other images, the direct use of the LiRa-features faces difficulties.

In the neural network with competitive layers there are no artificially and preliminarily generated features – the features are naturally formed during the learning process. These features are, in fact, pairwise combinations of one-valued pixels in all layers of the network. Creation of these features contributes to generalization and formation of adequate class descriptions in the network layers.

It is worth to note that it both classifiers (modular neural networks with the LiRA-features and neural network with competitive layers) the recognition rates and shapes of the experimental graphics are rather unstable. Namely, they are highly variable and strongly depend on a large number of different parameters, even those parameters that, at first glance, should not at all influence the results. For example, in the present experiments with the layer neural network, even a change in the order of distortions of the original shapes of the training characters in the learning procedure leads to a significant change in the shape of the curve of Fig. 3.

The presented neural network with competitive layers has a very simple structure and functioning mechanisms, so that it does not require any complex computations. We consider this to be an important advantage of the network. In addition, the functioning of the network has been further simplified, namely, the iterative multi-step procedure used in previous CLM-like networks ([2], [3]) has been replaced by a one-step calculation of neural activity in the recognition process. Consequently, the computer implementation of the network has acquired a high speed of the recognition procedure, which is a significant advantage for possible applications.

Experiments conducted in this work with the neural network with competitive layers confirm its effectiveness for the task of recognizing handwritten characters. The obtained results suggest application of this network to solve other tasks where data classification into a limited number of classes is required and a large number of training samples is available.

References

[1] H. Ritter, "A spatial approach to feature linking", *Proceedings of the International Neural Network Conference*, Paris, 2: 898-901, 1990. https://doi.org/10.1007/978-94-009-0643-3_124

[2] A. Goltsev, D. Rachkovskij, D. Wunsch, "A neural network for segmentation of line drawings into lines of different orientations", *Intelligent Engineering Systems through Artificial Neural Networks*, 10: 217-224, 2000. ASME PRESS, ISBN 0-7918-0161-6

[3] A. Goltsev, D. Rachkovskij, "A recurrent neural network for partitioning of hand drawn characters into strokes of different orientations", *International Journal of Neural Systems*, 11(5): 463-475, 2001. DOI:10.1142/S0129065701000862

[4] H. Wersing, J.J. Steil, H. Ritter, "A competitive-layer model for feature binding and sensory segmentation", *Neural Computation*, 13(2): 357-387, 2001. https://doi.org/10.1162/089976601300014574

[5] J. Ontrup, H. Ritter, "Perceptual grouping in a neural model: Reproducing human texture perception". In: *Technical Report SFB 360 98-6*, University of Bielefeld, 1998.

[6] B. Zheng Y. Zhang "Some properties of the competitive layer model with application to object regions extraction", *International Conference on Communications, Circuits and Systems*, Chengdu, China: 491-495, 2010. DOI: 10.1109/icccas.2010.558194

[7] B. Zheng, Z. Yi, "Using competitive layer model implemented by Lotka–Volterra recurrent neural networks for detecting brain activated regions from fMRI data", *Neural Computing and Applications*, 22: 395-404, 2013. https://doi.org/10.1007/s00521-012-0972-8

[8] B. Zheng, "A winner-take-all Lotka-Volterra recurrent neural network with only one winner in each row and each column", *Neural Computing and Applications*, 22: 1-9, 2013. DOI: 10.1007/s00521-013-1412-0

[9] A. Goltsev, V. Gritsenko, D. Húsek, "Segmentation of visual images by sequential extracting homogeneous texture areas", *Journal of Signal and Information Processing*, 11(4): 75-102, 2020. DOI: 10.4236/jsip.2020.114005

[10] V.K. Govindan, A.P. Shivaprasad, "Character recognition – a review", *Pattern Recognition*, 23(7): 671-683, 1990. https://doi:10.1016/0031-3203(90)90091-X.

[11] T. Gajjar, R. Teraiya, G. Gohil, M. Goyani, "Top down hierarchical histogram based approach for printed Devnagri script character isolation". In: Nagamalai D., Renault E., Dhanuskodi M. (eds) *Advances in Digital Image Processing and Information Technology. DPPR 2011. Communications in Computer and Information Science*, Springer, Berlin, Heidelberg: 205 (55-64), 2011. https://doi.org/10.1007/978-3-642-24055-3_6.

[12] P. Simard, Y. LeCun, J. Denker, "Efficient pattern recognition using a new transformation distance", *Advances in Neural Information Processing Systems*, 5: 50-58, 1993.

[13] A. Baldominos, Y. Saez, P. Isasi, "A survey of handwritten character recognition with MNIST and EMNIST", *Applied Sciences*, 9(15): 3169, 2019. https://doi.org/10.3390/app9153169

[14] K. Kowsari, M. Heidarysafa, D.E. Brown, K.J. Meimandi, L.E. Barnes, "RMDL: Random multimodel deep learning for classification", *Proceedings of the 2nd International Conference on Information System and Data Mining*, Lakeland FL USA: 19-28, 2018. https://doi.org/10.1145/3206098.3206111

[15] D. Krotov, J. Hopfield, "Unsupervised learning by competing hidden units", *Proceedings of the National Academy of Sciences of the USA*, 116(16): 7723-7731, 2019. DOI: 10.1073/pnas.1820458116

[16] http://yann.lecun.com/exdb/mnist/

[17] A. Goltsev, *Neural Networks with the Assembly Organization*, Kiev, Naukova Dumka, 2005 (in Russian). ISBN: 966-00-0520-2

[18] A. Goltsev, V. Gritsenko, "Investigation of efficient features for image recognition by neural networks", *Neural Networks*, 28: 15-23, 2012. DOI:10.1016/j.neunet.2011.12.002

[19] A. Goltsev, V. Gritsenko, "Modular neural networks with radial neural columnar architecture", *Biologically Inspired Cognitive Architectures*, 13: 63-74, 2015. https://doi.org/10.1016/j.bica.2015.06.001

[20] E. Kussul, T. Baidyk, "Improved method of handwritten digit recognition tested on MNIST database", *Proceedings of the 15-th international conference on vision interface*, Calgary, Canada: 192-197, 2002. https://doi.org/10.1016/j.imavis.2004.03.008

[21] E. Kussul, T. Baidyk, "LIRA neural classifier for handwritten digit recognition and visual controlled microassembly", *Neurocomputing*, 69(16-18): 2227–2235, 2006. https://doi.org/10.1016/j.neucom.2005.07.009

[22] Kussul, E., Baidyk, T., Wunsch, D. *Neural Networks and Micro Mechanics*, Springer–Verlag, 2010. DOI: 10.1007/978-3-642-02535-8

[23] D.O. Hebb, *The Organization of Behavior*, John Wiley & Sons Inc., 1949. DOI: 10.1016/s0361-9230(99)00182-3

[24] D.H. Hubel, T.N. Wiesel, "Brain mechanisms of vision", *Scientific American*, 241(3), 150-162, 1979. DOI: 10.1038/scientificamerican0979-150

[25] D.H. Hubel, *Eye, Brain and Vision*. New York, USA: Scientific American Library, 1988. ISBN: 0716750201