Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 1927–1934

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

Hierarchical Convolutional Neural Network for Handwritten Digits Recognition

Zufar Kayumov, Dmitrii Tumakov, Sergey Mosin*

Kazan Federal University, 18 Kremlyovskaya street, Kazan 420008, Russian Federation

Abstract

The application of a combination of convolutional neural networks for the recognition of handwritten digits is considered. Recognition is carried out by two sets of the networks following each other. The first neural network selects two digits with maximum activation functions. Depending on the winners, the next network is activated, which selects one digit from two. The proposed algorithm is tested on the data from MNIST. The minimal handwriting recognition error was estimated with this approach.

Keywords: Handwritten digitss; recognition; Hierarchical convolutional neural network; MNIST.

1. Introduction

Currently, neural networks play a significant role in our lives. Neural networks are widely used in science [1, 2], technology [3, 4] and many other fields [5, 6]. One of the main applications of networks is image processing. This includes various tasks related to the classification, localization and recognition of objects. Similar tasks are associated with the development of principles and the construction of systems designed to determine whether an object belongs to one of the classes of objects. Classes of objects can be pre-allocated (classification task) or it is

* Corresponding author. Tel.: +7-843-233-7037.
E-mail address: smosin@ieee.org

10.1016/j.procs.2020.04.206

necessary to identify them in the process of solving the problem (clustering problem). Under objects in pattern recognition, we understand various objects and phenomena, processes and situations, signals, etc. For example, neural networks are used to recognize objects and symbols [7], digits [8], as well as to recognize human actions [9], facial emotions [10, 11], and pedestrian detection [12].

The most common networks for this class of the tasks are convolutional neural networks [13, 14]. Here, thanks to the use of convolutional layers, the input data is filtered from unnecessary details. This allows further processing of only useful information, due to which there is an effective recognition of objects. The architecture of convolutional neural networks was firstly introduced in 1998 by Jan Lekun (LeNet architecture) [15, 16]. Special attention was paid to convolutional networks only after 14 years, when in 2012 the architecture of AlexNet [17] (modified and improved LeNet) won the ImageNet contest by a wide margin. Since then, convolutional neural networks have been actively used when working with visual data [18, 19].

In present work, we consider the recognition of handwritten digits from the MNIST database [20] by a hierarchical convolutional neural network that interprets the image iteratively, sequentially obtaining the correct answer [21]. Previously, various authors have repeatedly made numerous attempts to achieve maximum accuracy in recognizing handwritten digits. For example, when using a single-level perceptron, the error was 12%, and for two-layer networks using elastic deformations, the error reached to 0.7% [22]. However, the highest results were obtained only using deep convolutional neural networks. For example, a set of 35 six-layer convolutional neural networks with pre-processing and wide normalization for training showed 0.23% error [23], and when using an ensemble of five such networks with a significantly expanded data set, the error level was only 0.21% [24].

However, all of these methods are based on distortion and image pre-processing. Such methods involve increasing the data for training, due to which the sample becomes significantly larger, and the process of training takes a very long time. For example, in the case of elastic deformations, at the beginning of each era, the entire MNIST training kit is deformed. Affine transformations (rotation, scaling, and horizontal shift) occur over the images. It also requires a certain amount of time. In the hierarchical neural network that we have presented, all the above actions are not performed before training, and the added additional networks are quickly trained on small data sets.

The task is to test the possibilities of the presented approach with “average” training. Therefore, it is obvious that the results can be somewhat improved, for example, due to preprocessing [25], selection of activation functions [26], and other methods.

2. The first level. Digit recognition by convolutional neural network

We consider the architecture of a six-layer convolution network, consisting of an input layer containing 784 neurons (one image from the MNIST database has a dimension of 28 by 28 pixels) and six layers:

1. Convolutional layer, 11 cards of signs of dimension 24x24 (core bypass 5x5).
2. Subsample layer, 11 cards of signs of dimension 12x12.
3. Convolutional layer, 22 feature cards of dimension 8x8 (core bypass 5x5).
4. Subsample layer. 22 cards of signs of dimension 4x4.
5. Fully connected layer, 120 neurons.
6. Fully connected layer, 10 neurons.

We train the network using the backpropagation algorithm [27].

Note that the network parameters are established experimentally. For example, in this way the optimal digit of matrices in the first convolutional layer was obtained, containing 11 24-by-24 feature maps. The network was trained and tested on data representing 28 by 28 pixel handwritten digits from the MNIST database.

The smallest error of the convolutional neural network shown in Fig. 1, in the test sample from MNIST it amounted to 0.82% (82 images from the test sample containing 10 thousand handwritten images were incorrectly recognized). Note that we did not achieve maximum accuracy at this stage.

We write the digits of incorrectly recognized MNIST images: 115, 247, 259, 449, 582, 619, 659, 674, 740, 791, 900, 947, 1014, 1039, 1112, 1226, 1232, 1247, 1260, 1299, 1319, 1364, 1393, 1414, 1522, 1527, 1549, 1621, 1709, 1901, 2035, 2070, 2109, 2130, 2135, 2293, 2414, 2447, 2488, 2597, 2654, 2921, 2927, 3225, 3330, 3422, 3503,

3520, 3597, 3767, 3780, 3808, 3941, 3985, 4078, 4176, 4248, 4369, 4571, 4807, 4874, 4956, 5752, 5937, 5955, 5973, 6532, 6576, 6597, 6651, 6783, 7216, 8246, 8316, 9009, 9634, 9664, 9679, 9692, 9698, 9729, 9770.

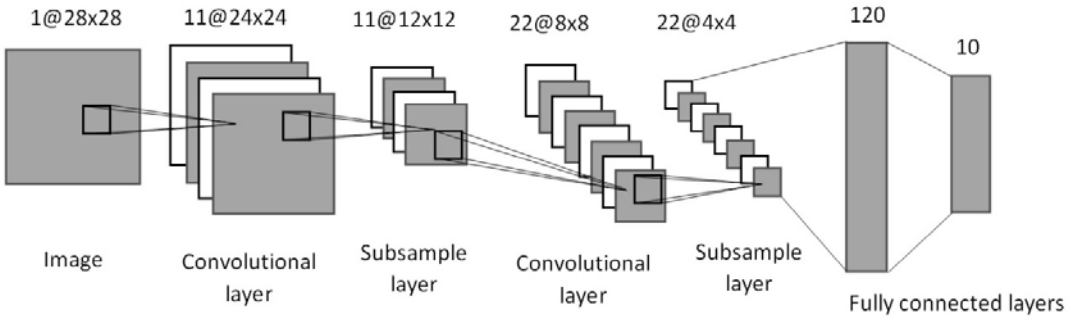

Fig. 1. Convolutional Neural Network Architecture.

Without dwelling on each of the incorrectly recognized images in detail, we present a general error table for the convolution network (Table 1). From the table we can see which digits and how many times the neural network was confused during recognition. For example, when testing, we found that the neural network instead of the digit 9 received the digit 4 five times, and four times the output instead of the digit 4 received 9. In total, nine errors occurred between the four and nine. This value is listed in the table. We also see that a large digit of incorrectly issued results were received between the three and the five: 8 errors.

Table 1. Cross-table of errors for the convolutional network.

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	-	-	2	4	2	3	1
1	-	-	1	1	-	1	5	2	1	1
2	-	-	-	2	4	-	-	6	3	1
3	-	-	-	-	-	8	-	1	5	2
4	-	-	-	-	-	-	3	-	1	9
5	-	-	-	-	-	-	2	1	-	3
6	-	-	-	-	-	-	-	-	1	-
7	-	-	-	-	-	-	-	-	1	5
8	-	-	-	-	-	-	-	-	-	1
9	-	-	-	-	-	-	-	-	-	-

Let us demonstrate the “successful” recognition of a digit from a test sample of the MNIST database. We consider, for example, the 102nd image of MNIST (the left image in Fig. 2). The network fairly confidently gives the activation value 0.999684 for the digit 5. The remaining output neurons in this case have values less than 0.0008; the closest of them is the digit 3 (the value at the output is 0.00078).

We consider now an example of an incorrectly recognized digit 7 (the 1260th digit MNIST, the central image in Fig. 2). The activation of the digit 1 is 0.62637, and the activation of the 7 is 0.14691. The activation difference is 0.47946. This indicates the "uncertainty" of the neural network in the correctness of the output. We note that the digit 2 has an activation value of 0.01, while the remaining digits are of the third order after a point or less.

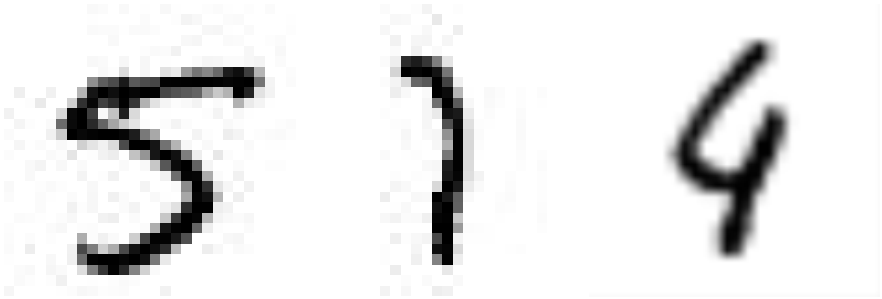

Fig. 2. Images digit 102, 1260 and 115 from the MNIST.

In the given example, the “correct” digit had the second largest activation value. Such a situation with erroneous recognition is quite common. In this regard, it can be assumed that an improvement in recognition can be achieved by creating an additional network that will be trained to recognize two digits among themselves.

3. The second level. Additional convolutional networks

Additional networks are a set of convolutional neural networks trained to recognize two digits. We have analyzed Table 1 and found out that 31 such networks are needed (equal to the digit of nonzero cells in the table).

The recognition algorithm using a combination of neural networks has the following two stages:

1) Passage through the global network;

2) Passage through the additional network if the difference between the maximum values of the digits at the output does not exceed 0.7 at the first stage (the value of 0.7 was obtained experimentally).

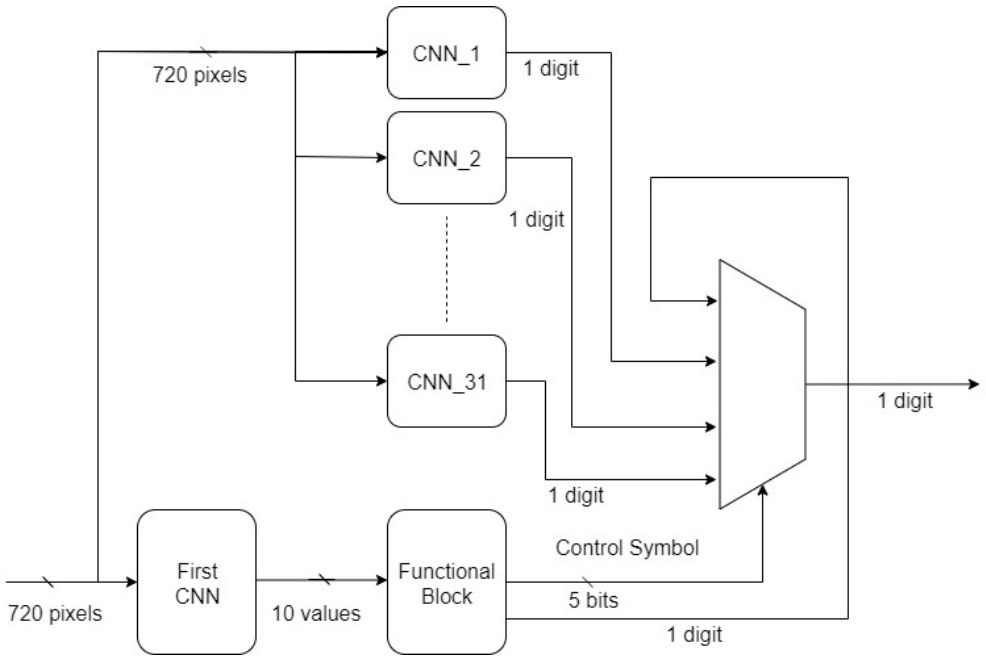

Fig. 3. Scheme of the hierarchical neural network.

From a technical point of view, we present the general architecture of our multi-level network in a diagram (see Fig. 3). The input (on the left) is an image consisting of 720 pixels. This data enters the first neural network

described in point 2. Also, the data is fed to the inputs of the 31 neural network of the second level, shown in the diagram at the top.

At the output of the first neural network, we get ten values (from zero to one) for each digit. Further, these values enter the function block. The function block analyzes the ten values obtained: if the difference in values for the winning neuron and the "second place" neuron is greater than 0.7, then a five-bit control symbol with all zeros is generated; if the difference in values is less than 0.7, then a code is generated corresponding to the "necessary" neural network from the second level. Also, the winner figure comes out of the function block separately.

Each of the 31 convolutional neural networks of the second level has almost the same architecture as the network of the first level. The only difference is that at the output of the second-level network there are two neurons instead of ten (see Fig. 1). Second-level networks choose one digit out of two. For example, CNN_1 chooses between 0 and 5, and this network is trained on a sample containing only these numbers.

Thus, the last block includes 32 winning digits: 1 digit from the first-level network and 31 digits from the second-level network, as well as a control character indicating which neural network to select the digit from (it will be at the output of this block).

The error matrix is presented in Table 2. Comparing tables 1 and 2, we conclude that the errors between the digits 0-5, 1-3, 1-5, 2-3, and 4-8 are completely removed, and the recognition of pairs 1-6, 2-4, 2-8, 3-8, 4-9 and 7-9 is improved. For example, we seriously managed to improve recognition between 4 and 9: out of nine errors, five remained. As a result, with the recognition of digits, 61 incorrectly defined images remain.

Table 2. Confusion matrix of additional networks.

	0	1	2	3	4	5	6	7	8	9
0	973	-	-	-	-	1	2	2	2	-
1	-	1131	-	1	-	1	1	-	1	-
2	-	1	1026	1	1	-	-	2	1	-
3	-	-	-	1004	-	2	-	-	4	-
4	-	-	3	-	973	-	2	-	-	4
5	1	-	-	6	-	882	1	1	-	1
6	2	4	-	-	1	1	949	-	1	-
7	-	2	4	1	-	-	-	1017	1	3
8	1	-	2	1	1	-	-	-	968	1
9	1	1	1	2	5	2	-	2	-	995

We consider the example of the 115th digit from MNIST correctly recognized in the second stage (the right digit in Fig. 2). The first stage produces the following result: the value at the output of the digit 9 is 0.79, and that of digit 4 is 0.35. The activation difference is 0.44 in favor of 9. Since the activation difference is less than 0.7, we pass the image through an additional network that differs 4 from 9. We get 0.87 for the digit 4 and 0.1 for the 9.

All the digits successfully corrected by the additional network are shown in Fig. 4. The following image digits from the MNIST database correspond to them: 115, 582, 740, 900, 1364, 1414, 1522, 1527, 2070, 2447, 2927, 3330, 3941, 3985, 4176, 4956, 5752, 5973, 8316, 9679, 9698, 9770.

As a result, 64 incorrectly recognized images remained at the output of neural networks. Their digits are the following: 247, 259, 449, 619, 659, 674, 791, 947, 1014, 1039, 1112, 1226, 1232, 1247, 1260, 1299, 1319, 1393, 1549, 1621, 1709, 1737, 1754, 1901, 2035, 2109, 2130, 2135, 2293, 2414, 2488, 2597, 2654, 2921, 3225, 3422, 3503, 3520, 3597, 3767, 3780, 3808, 4078, 4248, 4369, 4571, 4807, 4874, 5937, 5955, 6532, 6576, 6597, 6651, 6783, 7216, 8246, 8527, 9009, 9015, 9634, 9664, 9692, 9729.

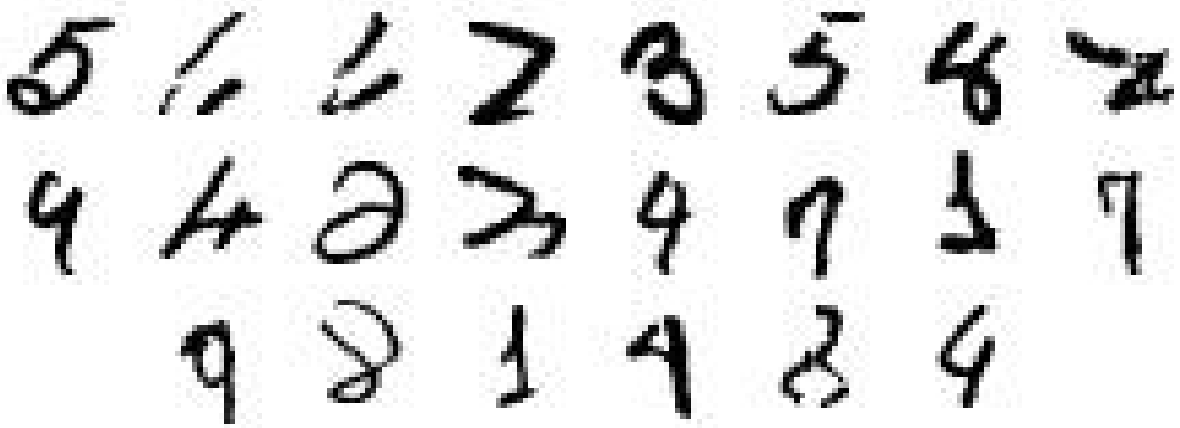

Fig. 4. Images not recognized at the first stage, but determined at the second.

Thus, the digit of correct digits is 9936 out of 10000, and the recognition error using the obtained networks is 0.64%. The traditional F-metric was considered also. Table 3 shows the accuracy, completeness, and F-metric for each of the ten digits.

Table 3. Table of recognition errors of individual digits.

Metric	0	1	2	3	4	5	6	7	8	9
Recall	0.995	0.993	0.990	0.988	0.992	0.992	0.994	0.993	0.990	0.991
Precision	0.993	0.996	0.994	0.994	0.991	0.989	0.991	0.989	0.994	0.986
F-measure	0.994	0.995	0.992	0.991	0.991	0.990	0.992	0.991	0.992	0.989

The table shows that all digits have the F-measure of the order of 0.99. We can conclude that the “worst” digit for recognition is 9 (F = 0.989), and the “best” one is 1 (F = 0.995).

4. Error estimation of the received convolutional neural networks

The minimum error of the proposed algorithm was calculated. The results of the first stage for test images were evaluated for this purpose. Two digits with the largest activation values were selected at the output of the first network. If none of the two remaining digits is correct, then the additional network cannot provide the correct value. And, obviously, the digit of errors on the first network is the smallest possible error.

As a result, we received 21 such images: 247, 1014, 1039, 1226, 1232, 1247, 1709, 2109, 2135, 2293, 2921, 3225, 3520, 3780, 3808, 4078, 4248, 4571, 4807, 5955, 8246. All of these images are presented in Fig. 5. Thus, the maximum accuracy that can be achieved using the presented algorithm is 99.79%.

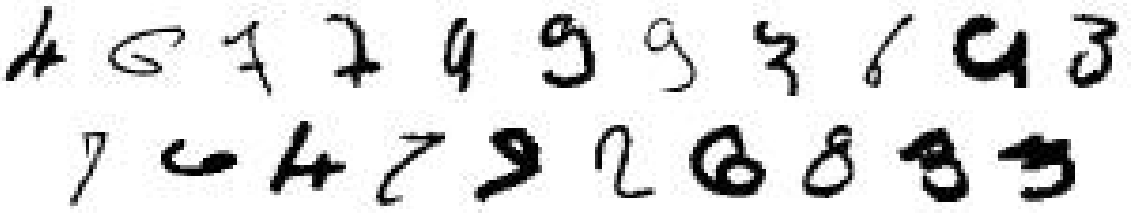

Fig. 5. Images that did not pass the first stage (did not fall into the pair of winners).

Note that digits shown in Fig. 5 do not fall into the pair of winners of the first stage, but they have close output values. Indeed, for the first digit in Fig. 5, which is the 247th digit from the MNIST test sample, the correct value is 4. However, at the first stage, the coefficients 0.722 and 0.223 were obtained for digits 2 and 6 correspondingly. For the correct digit 4 the result is only 0.03, and for the remaining digits the values are much smaller.

5. Conclusions

The hierarchical convolutional neural network is proposed for the task of recognizing handwritten numbers. The similar task can be used for handwriting recognition on tablet computers, recognition of postal codes for sorting mail, processing bank checks, entering numerical entries in manually filled forms, etc.

It takes a lot of time to train classical neural networks for pattern recognition from large databases. In this paper, we consider an algorithm that spends a short time on training, but obtains good (refined) results through the use of auxiliary networks of the "second" level. A similar approach can be implemented in parallel implementation using SIMD technology.

The proposed algorithm is tested on data from MNIST. The recognition error was 0.64%. It is shown that the minimum error with this approach is 0.21%, and the accuracy of the F-measure is about 0.99 for each digit.

Acknowledgements

The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

References

[1] Tumakov, D.N., Khairullina, D.M., and Valeeva, A.A. (2017) “Recovery of parameters of a homogeneous elastic layer using neural networks.” *Journal of Fundamental and Applied Sciences* **9**: 1202–1220.

[2] Wachinger, C., Reuter, M., and Klein T. (2018) “DeepNAT: Deep convolutional neural network for segmenting neuroanatomy.” *NeuroImage* **170**: 434–445.

[3] Dautov, R., and Mosin, S. (2018) “Technique to aggregate classes of analog fault diagnostic data based on association rule mining.” in *Proceedings of 19th International Symposium on Quality Electronic Design:* 238–243.

[4] Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018) “ShuffleNet: An extremely efficient convolutional neural network for mobile devices.” in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*: 6848–6856

[5] Dreyfus, G. (2005) *Neural Networks Methodology and Applications*, Springer-Verla.

[6] Veelenturf, L. P. J. (1995) *Analysis and Applications of Artificial Neural Networks*, Prentice Hall.

[7] Neha, S., Vibhor, J., and Anju, M. (2018) “An analysis of convolutional neural networks for image classification.” *Procedia Computer Science* **132**: 377–384.

[8] Shruti, R., Kulkarni, and Bipin, R. (2018) “Spiking neural networks for handwritten digit recognition – Supervised learning and network optimization.” *Neural Networks* **103**: 118–127.

[9] Earnest, P.I., and Krishna, M.C. (2016) “Human action recognition using genetic algorithms and convolutional neural networks.” *Pattern* **59**: 199–212.

[10] Xuanyu, H., and Wei, Z. (2018) “Emotion recognition by assisted learning with convolutional neural networks.” *Neurocomputing* **291**: 187–194.

[11] Mesbah, A., Berrahou, A., Hammouchi, H., Berbia, H., and Daoudi, M. (2019) “Lip reading with hahn convolutional neural networks.” *Image and Vision Computing* **88**: 76–83.

[12] Tomè, D., Monti, F., Baroffio, L., Bondi, L., and Tubaro, S. (2016) “Deep convolutional neural networks for pedestrian detection.” *Signal Processing: Image Communication* **47**: 482–489.

[13] Zeiler, M., and Fergus, R. (2014) “Visualizing and understanding convolutional networks.” in *Proceedings of the European Conference on Computer Vision*: 818–833.

[14] Schmidhuber, J. (2015) “Deep learning in neural networks: An overview.” *Neural Networks* **61**: 85–117.

[15] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998) "Gradient-based learning applied to document recognition", in *Proceedings of the IEEE* **86**: 2278–2324.

[16] Convolutional Neural Networks (LeNet) – *DeepLearning 0.1 documentation*. DeepLearning 0.1. LISA Lab.

[17] Krizhevsky, A. (2012) “ImageNet: Classification with deep convolutional neural networks.” in *Proceedings of the 25th International Conference on Neural Information Processing Systems* **1**: 1097–1105.

[18] Neha, J., Shishir, K., Amit, K., Pourya, S., and Masoumeh, Z. (2018) "Hybrid deep neural networks for face emotion." *Pattern Recognition Letters* **115**: 101–106.

[19] Jiuxiang, G., Zhenhua, W., Jason, K., Lianyang, M., and Tsuhan, C. (2018) "Recent advances in convolutional neural networks." *Pattern Recognition* **77**: 354–377.

[20] The MNIST database handwritten digits. URL: http://yann.lecun.com/exdb/mnist.

[21] Behnke, S. (2003) *Hierarchical Neural Networks for Image Interpretation*, Springer-Verlag.

[22] Simard, P., Dave, S., and Platt, J. (2003) "Best practices for convolutional neural networks applied to visual document analysis." in *Proceedings on Seventh International Conference*: 958–962.

[23] Cires, D., Ueli, M., and Jürgen, S. (2012) "Multi-column deep neural networks for image classification." in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*: 3642–3649.

[24] Romanuke, V. (2016) "Training data expansion and boosting of convolutional neural networks for reducing the MNIST dataset error rate." *Research Bulletin of NTUU "Kyiv Polytechnic Institute"* **6**: 29–34.

[25] Mosin, S. (2019) "Machine learning and data mining methods in testing and diagnostics of analog and mixed-signal integrated circuits: Case study." *Communications in Computer and Information Science* **968**: 240–255.

[26] Latypova, R., and Tumakov, D. (2018) "Method of selecting an optimal activation function in perceptron for recognition of simple objects." in *Proceedings of 16th IEEE East-West Design and Test Symposium*: 390–394.

[27] Hecht-Nielsen, R. (1992) "Theory of the backpropagation neural network." *Neural Networks* **2**: 65–93.