Fuzzy Logic Module of Convolutional Neural Network for Handwritten Digits Recognition

E.A. Popko and I.A. Weinstein

NANOTECH Centre, Ural Federal University, Mira Street, 19, Yekaterinburg, Russia

e.a.popko@urfu.ru

Abstract. Optical character recognition is one of the important issues in the field of pattern recognition. This paper presents a method for recognizing handwritten digits based on the modeling of convolutional neural network. The integrated fuzzy logic module based on a structural approach was developed. Used system architecture adjusted the output of the neural network to improve quality of symbol identification. It was shown that proposed algorithm was flexible and high recognition rate of 99.23% was achieved.

1. Introduction

Fuzzy logic is actively used in many technological problems related to the pattern recognition [1]. In particular, great attention is paid to accurate identification of handwritten characters in classification and analysis of documents, automatic serial number recording, and vehicle identification[1, 2]. One of the promising trends is the employment of artificial convolutional neural networks to imitate biological nervous systems with learning ability.

Convolutional neural networks are the extension of a multi-layer perceptron, optimized for two-dimensional pattern recognition (a character identification task), face detection, and so on[3]. The existence of neural layers of different types and functionality is a distinguishing attribute of CNNs. CNNs combine three architectural ideas: local receptive fields to extract elementary features from images; shared weights to extract the same set of elementary features from the entire input image and to lower computational costs; local averaging and sub-sampling to reduce the resolution of feature maps. The goal of this work is to recognize handwritten digits and improve the classification results by way of a newly devised version of convolutional neural network through integrating a logical correction module into their structure.

2. Model

Traditional character recognition methods can be divided into three main types - template, structural and feature ones [4, 5], each of which has its own advantages and restrictions (Table 1). At present, it can be boldly argued that the most rapidly developing algorithms are those of classification, including the strategy of learning by examples. Convolutional neural networks also possess the aforementioned peculiarity. As distinct from a classical multilayer perceptron, CNNs provide the invariance of image analysis to shift, rotation, scale, and distortions.

Table 1. The main classes of the recognition methods

Methods	Advantages	Disadvantages
Template	High speed, ability to work under existence of character defects, provides noise immunity	Weak versatility, fails to identify the font, which differs from the input
Structural	It doesn't matter what size and font are	Complexity of defect-character recognition, works relatively slow
Feature	Good generalizing ability, resistance to changes in the shape of characters	Instability to various defects, loss of information part

In this work, as a prototype, we have used the network architecture described in [6]. The main difference of the implementation is to use six feature maps in a second layer, instead of five. The general functional scheme of CNN is shown in Figure 1. The input layer receives images of recognizable characters. This is followed by a layer of six convolutional feature maps of size 13 × 13. Each map element is connected to the receptive field of 5 × 5 on an input image. Further, there is an additional convolutional layer consisting of 50 maps of size 5 × 5. The next layer is fully-connected and contains 100 neurons. The fifth layer contains neurons corresponding to the digits from 0 to 9.

Figure 1. The neural network architecture being used
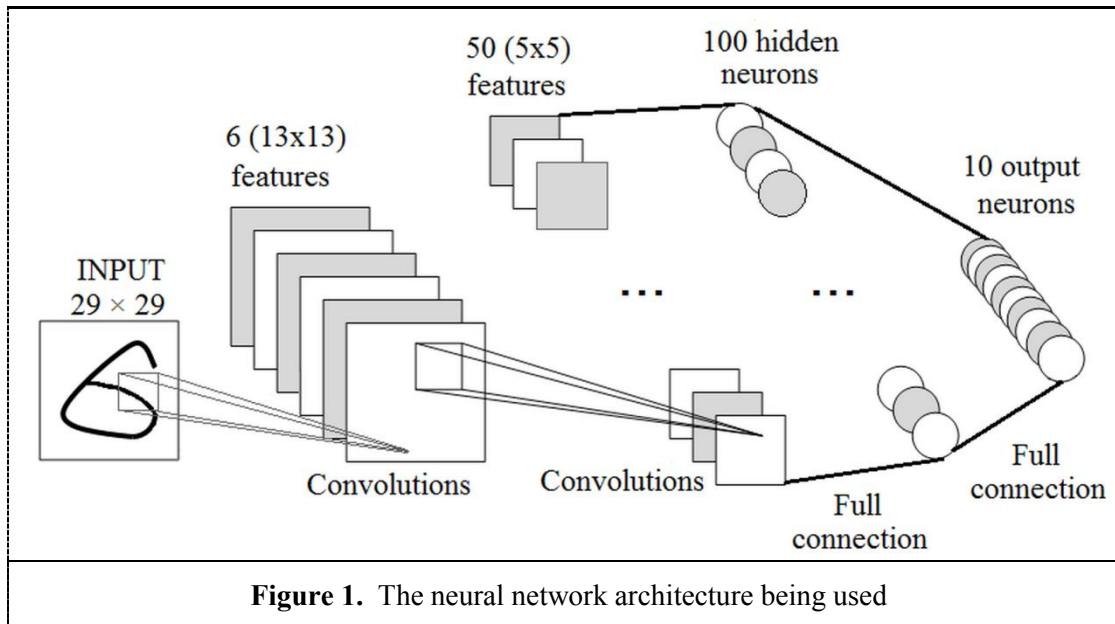
The connection weights of the convolutional neurons located in one feature map are shared, and the position of the local feature becomes less important. Such a way allows the shift invariance to be achieved. The output neural values Y_n^l of the n-th feature map of the convolutional layer l can be calculated as follows [3]:

$$Y_n^l(x, y) = f\left(\sum_{m \in M_n^l} \sum_{i=0}^{K^l-1} \sum_{j=0}^{K^l-1} \omega_{mn}^l(i, j) \cdot Y_m^{l-1}(x+i, y+i) + b_n^l \right), \tag{1}$$

where x, y are coordinates of a neuron inside the feature map; M_n^l is the set of feature maps of the previous layer l–1 that are associated with the n-th feature map of the layer l; ω_{mn}^l is the matrix of synaptic coefficients (the convolution kernel); K^l is the size of the receptive field of the neurons of the l-th layer; b_n^l being the bias term for the n-th feature map of the l-th layer.

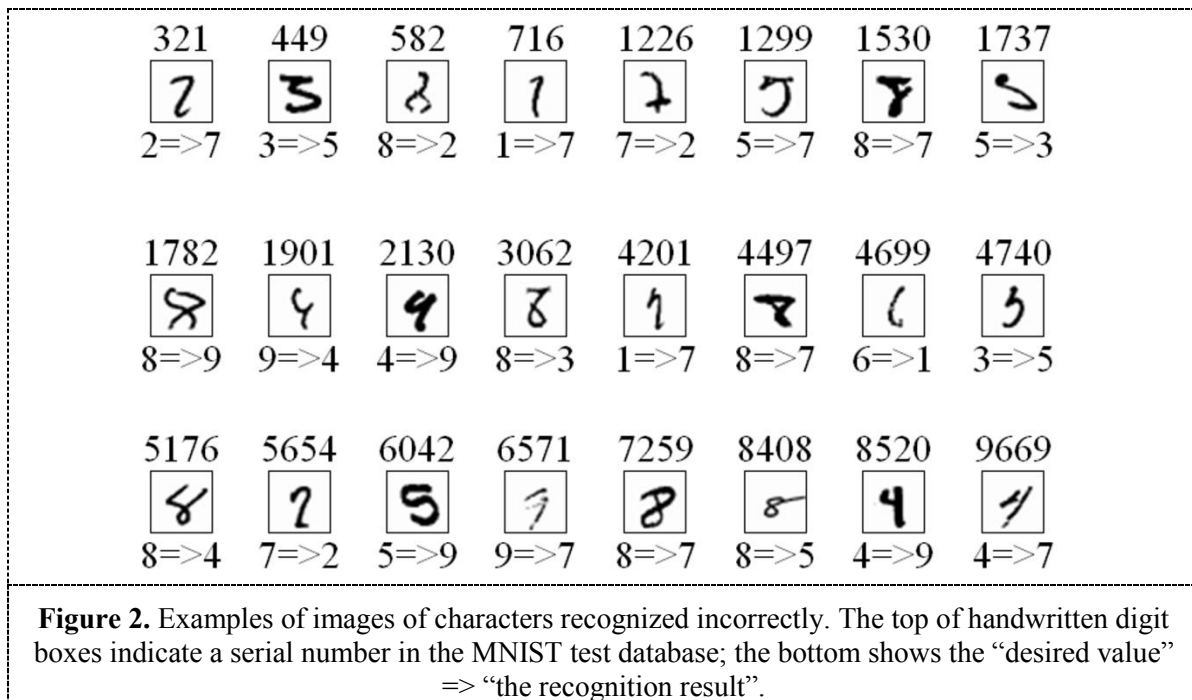As an activation function $f(z)$ of the neural network, the scaled hyperbolic tangent is used here:

$$f(z) = A \cdot \tanh(Sz),\qquad\qquad(2)$$

where z is an input value corresponding to the neuron activation value; A and S are parameters of the function. The work [3] proposes to employ the values of $A = 1.7159$ and $S = 2/3$.

This paper is based on the standard back-propagation algorithm to tune the neural network. As data for training and testing, we have utilized a set of images of the MNIST handwritten digit database [7].

3. Algorithm

As shown by preliminary tests, the major difficulty of CNNs is to identify gaps in specific areas of characters and individual structural elements, for the classification. The network errors mentioned above appear due to the omission of these features (Figure 2). For example, in the case when the character "2" was incorrectly identified as "7", the neural network was not able to determine clearly whether the bottom element as a line corresponds to two or seven (see the sample 321 in Figure 2).

Figure 2. Examples of images of characters recognized incorrectly. The top of handwritten digit boxes indicate a serial number in the MNIST test database; the bottom shows the "desired value" => "the recognition result".

We have introduced a fuzzy logic module into the system. Its main intention is to ascertain the existence of certain structural elements of recognizable characters and to yield an appropriate correction penalty for the neural network outputs. The logical module includes a rule database for producing correction values. Each fuzzy rule can be represented as a vector model: $P = <Name, Alg, Rule, Outs, Val>$, where *Name* is the name of the rule; *Alg* is the algorithm for ascertaining the existence of the element; *Rule* is the definition of fuzzy rule; *Outs* is the set of the neural network outputs being corrected; *Val* is the maximum correction value.

For example, for more reliable identification of the characters "2" and "7" , the rule is formulated as follows: P_1 = <"correction for the character 7", "algorithm that determines the correspondence of the lower element (the line) to the digit "7" or "2", "if (element is wide) then apply penalty", {7}, -0.5>. In this case, the penalty -0.5 is assigned to the network output neuron, which is responsible for the digit "7" provided that the bottom of the symbol has a horizontal underline of sufficient size, corresponding to the digit "2". The corresponding membership function μ is shown in figure 3.

Figure 3. Example of fuzzy membership function for the characters "7" and "2"
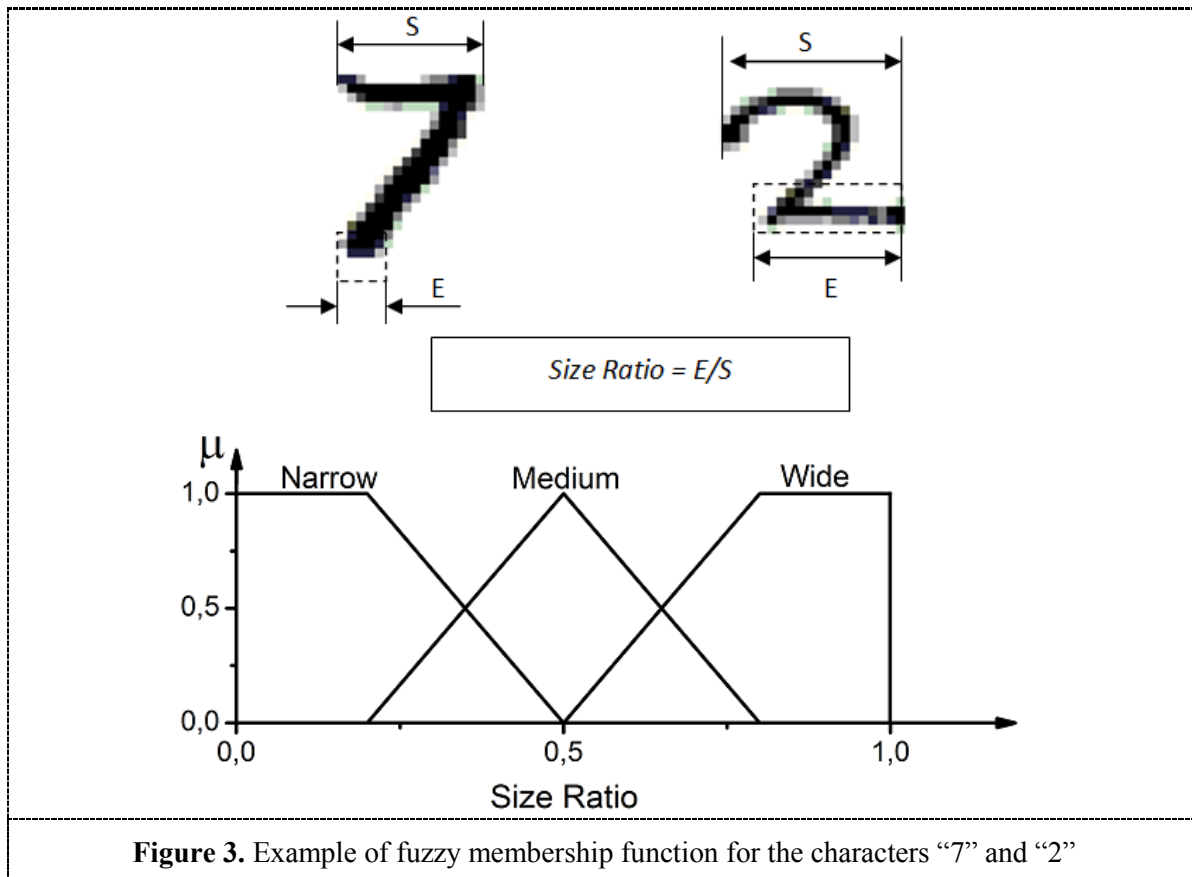
4. Results
For testing, we have devised a software module in C++ with an open source cross-platform computer vision library (OpenCV) to implement the recognition method proposed [8]. Figure 4 displays the overall scheme of verifying the system's performance capability.

Figure 4. The system testing scheme
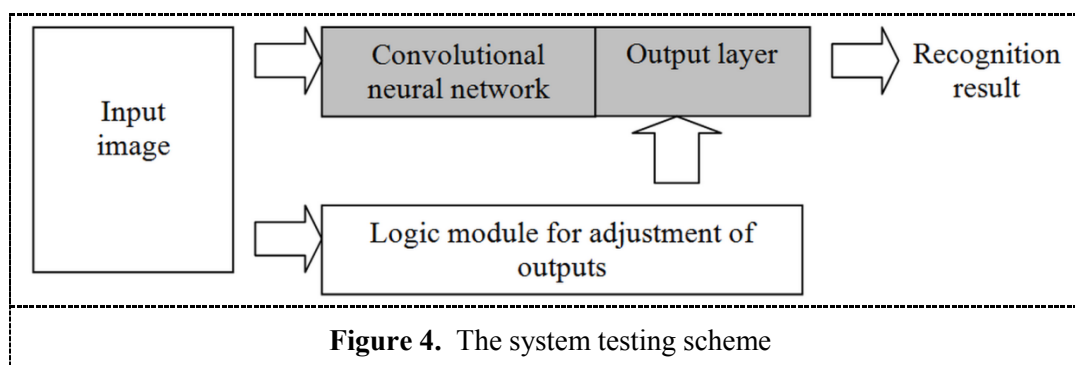
The verification of the system was first performed on a training set of 60000 characters, and then run on a test one of 10000 examples of the MNIST base. The training of the network was held for 50 epochs, the total tuning time amounted to 8 hours. The program was governed using both the logic correction module and without it. The findings of both program implementations are reported in Table 2.

Table 2. Recognition results

Implementation	Recognition accuracy for the training set	Recognition accuracy for the test set
Original CNN	97.35%	99.17%
Using the correction module	99.10%	99.23%

During the testing, it was found that the system equipped with the correction module allows the recognition quality to be enhanced. Over against the original CNN, the system succeeded in recognizing correctly digits from the test set with the numbers 321, 1299, 1530, 5654, 7259, 9669 (see. Figure 2). The achievable accuracy of the developed hybrid system is 99.23%, which is quite a good result [9, 10, 11]. Given the confident recognition, it can be claimed that one of the advantages of the proposed implementation is quite simple architecture of CNN as compared to analogs [9, 10]. The tuning of the configuration and internal network parameters for each particular case is a separate optimization problem. The traditional approach, as a rule, involves a series of preliminary experiments and the choice of structure. One of the most promising approaches uses evolutionary algorithms [12, 13, 14, 15].

5. Conclusion
Thus, in this study we have developed a modification of convolutional neural network using a fuzzy logic correction module. In this case, the embedded unit requires no implementation of a full recognition procedure, but merely modifies the network outputs by computer vision methods. It is shown that this simplified system architecture allows improving the quality of digit recognition and increases the accuracy to 99.23%.

The application of the convolution neural networks is quite a powerful tool for handwriting recognition owing to the possibility of their integrating into hybrid computer vision systems. Further investigations may be aimed at creating a learning technique of CNNs, advancing their optimal structure and fine-tuning of appropriate weights.

References
[1]   Ross T J 2010 *Fuzzy logic with engineering applications* (West Sussex, UK: John Wiley & Sons)
[2]   Latoria A, Chauhan A S, Saxena A S 2012 *International J. of Computers&Technology* **3** 458
[3]   LeCun Y 1999 *Proc. of IEEE* vol 86 2278
[4]   Davies E R 2012 *Computer and Machine Vision: Theory, Algorithms, Practicalities* (Oxford, UK: Elsevier)
[5]   Bishop C M 2006 *Pattern Recognition and Machine Learning* (New York: Springer)
[6]   Simard P Y, Steinkraus D, Platt J C 2003 *Proc. 7th Int. Conf. on Document Analysis and Recognition* p 958
[7]   The MNIST Database. Available from: http://yann.lecun.com/exdb/mnist/

[8]   Popko E A, Weinstein I A 2016 *Submitted certificate on official registration of the computer program* (Moscow, Russia)
[9]   Yang Z, Moczulski M, Denil M, et. al. 2015 *Proc. of the IEEE Int. Conf. on Computer Vision* p1476
[10] Lee H, Grosse R, Ranganath R, Ng A Y 2009 *ICML Proc. of the 26th Annual Int. Conf. on Machine Learning* p 609
[11] Thom M, Palm G 2013 *J. of Machine Learning Research* **14** 1091
[12] Oulette R, Browne M, Hirasawa K 2004 *Proc. of the 2004 IEEE Congress on Evolutionary Computation* **1** p 516
[13] Stanley K O, Miikkulainen R 2002 *Evolutionary Computation* **10** 99
[14] Weinstein I A, Popko E A 2008 *Radiat. Meas.* **43** 218
[15] Weinstein I A, Popko E A 2006 *Tech. Phys. Letters* **32** 534