OCR systems based on convolutional neocognitron network

Palka Jan, Palka Jiri, Navratil Milan

***Abstract*—** This paper deals with the recognition of handwritten text. It is mainly discussing improving nowadays OCR systems. In detail is this article focused on the possibilities of implementing the neocognitron network in this improvement. Next part deals with the problems of document processing, recognition of individual characters and subsequent search for whole words against the dictionary. Main goal of this work is to invent new principles in the field of processing hand written text especially focused on text with language specifics like diacritics.

***Keywords*—** OCR, MNIST, recognition, hand-written text, neural network.

I. Introduction

The core of the most OCR systems are, algorithms based on neural networks, hidden Markov models, minimum distance classifiers or support vector. These methods usually have a decent success rate in recognition of machine-written data. The handwritten data recognition is more complicated due to the variability of individual manuscripts. The best method for recognizing handwritten characters seems the neural network. There are many different structured neural networks, so I've chosen for further testing the multilayer neural network with error back spread. As the training set suitable for testing and verification of network capacity, for recognizing and classify a pattern was chosen database MNIST.

II. MNIST Database

This is the database of handwritten digits, containing 60,000 patterns for training and 10 000 patterns for testing purposes. These patterns were obtained from approximately 250 different authors. The database is suitable for testing learning algorithms for pattern recognition and classification (handwritten digits). Numbers are normalized in size and they are centered. The advantage in this database is because the patterns in the database do not require pre-processing or other treatment. Individual patterns in the database are stored as grayscale image, which is then normalized to 20x20 pixel size while maintaining aspect ratio. Because of anti-aliasing technique the image is transformed into shades of gray after normalization.

Fig 1. Sample of standard form for gathering data filled with language specific data

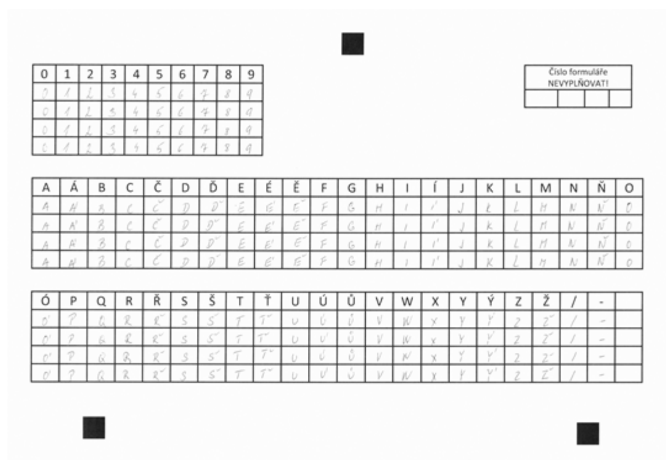

Fig. 2 Improved form used for gathering test data

The final operation is centered on the center of gravity image of the final image size of 28x28 pixels. The Matlab algorithm was developed to retrieve all the patterns from the database, their necessary adjustments and so prepared samples represented the input data for training and testing neural networks.

III. Multi-layer neural network (perceptron)

Multi-layer neural network is composed of at least three layers of neurons: input, output and at least one inner layer. Neurons in the output and the inner layer have a defined threshold, which corresponds to the weight value assigned by the connection between the neuron and a fictitious neuron, whose activation is always 1 between two adjacent layers of neurons occurs of full interconnection of neurons, each neuron in lower layer is connected to all neurons higher layers.

For learning neural network is necessary to have a training set containing elements describing the problem being solved and a method that can fix these patterns in a neural network by the values of synaptic weights. Each training set pattern describes how neurons are excited in the input and output layers.

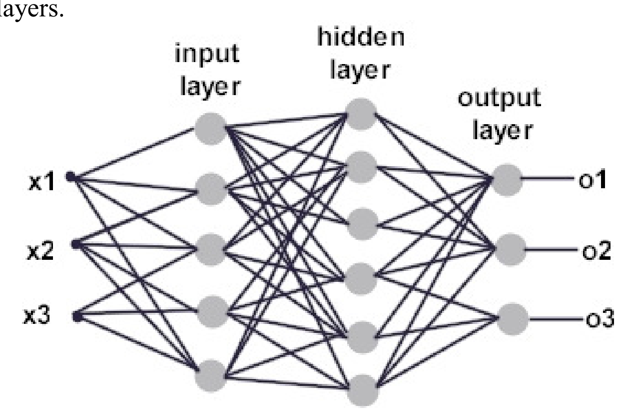

Fig 3. Multi-layer neural network

The most common adaptation algorithm of multilayer neural networks is the backpropagation method, which allows adaptation over the neural network training set. The actual algorithm consists of three stages: a forward spread of the input signal, reversing the spread of abnormality and updating the weight values on connections. During the forward spread, the signal is received by each neuron in input layer, and mediates its transfer to all neurons in the inner layer. Each neuron in the inner layer calculates its activation, and sends a signal to all neurons in output layer. Each neuron in the output layer calculates its activation, which corresponds to its actual output n-th neuron after the presentation of input pattern.

Experimentally it was found that it is not necessary to increase the number of hidden layers to values greater than 3. Specifying the number of neurons in the hidden layers is a classic problem. The idea that a larger number of neurons provide a greater percentage is misleading. With the increasing number of neurons in the hidden layer (or as the number of hidden layers) increases the nonlinear behavior of the network, but also a growing demand for learning (number of patterns, learning time). Too large network tends to be overfitting, etc. Too much focus on unimportant details occurring in the training set can happen, but these details may not are relevant for the resolution of problems. There is no general recommendation on how to choose the number of neurons. One of the heuristics used says that the number of neurons in the hidden layer should be twice the number of neurons in the input layer. Practically it is necessary to try multiple networks with different characteristics (topology, transition functions, etc.) and based on their behavior to choose the one that provides the best results.

IV. Testing of designed perceptron network

The Matlab algorithm was developed that performs multi-layer neural network. It is possible to choose the number of hidden layers, number of neurons in the hidden layers, types of transfer functions, adjust the methods of learning and many other parameters.

There were created two different types of neural networks. The first included a single network with number of output neurons corresponding to the number of classification groups. The second set featured a mesh of simple networks, which have only one output. The number of combinations needed was given by , where n is the number of characters. The proposed model was compared across a set of subnets. This method was much more time consuming to learn, especially when large numbers of characters.

A key attribute of the character recognition using neural networks is actually present on the network input (feature extraction). The main task of the extraction is to obtain a set of characteristics that maximize the success of classification and which will have the minimum possible number of elements. During the solution to this problem has been progressively tested various options, covering both statistical (zoning, design, sections, zero-crossing) and structural (histograms, Fourier transform) characteristics were tested and different torque characteristics. I started from the simplest, it means using conventional 1-bit image representation, the case of representing the feature vector by 0 or 1 according to the model and then continued with gradual adaptation of the model. It has taken a number of test measurements in order to find a combination that would be most appropriate in terms of correct recognition. An important feature of the proposed network was the ability to learn and correctly classify unknown patterns. In this case, the total network learned fairly, in presenting models from training sets were successfully classified over 90%, but after the presentation of unfamiliar patterns (the test set), this percentage dropped below 50%, which was totally unsatisfactory.

Further testing found that the network is incorrectly classified as small as a result of input pattern such as displacement, angle, scale, or different rotation. To eliminate such commonly occurring phenomena has been created an algorithm that removed the gradient and normalized in some way presented patterns. Success is then improved only a few units percent, but due to time expensivity of algorithms and the fact that only numbers were tested, so only classification groups, This approach was rejected. [1]

V. Convolutional neural network neocognitron

It is a multi-layered hierarchical neural network. Its advantage is the ability to correctly identify not only learned images but also images that arise in partly moving, rotating or other deformations. This hierarchy is based on that the network will detect lower levels of simple signs. With each level these symptoms are more and more complex. Each level always contains three layers: the S-layer, the C-layer V-layer. The only exception is the zero level, containing only the input layer, which stores the input information. The individual layers consist of a number of areas which are under the layers of either S, C or V. The area consists of two-dimensional array of cells. Neocognitron network consists of four basic types of cells: S-cells, C-cells, V-cells and receptor cells.

The basic elements, individual cells, have been working with real non-negative values. Neocognitron networks are characterized by a high density of connections between individual cells. Each cell is linked to a group of cells by attachment area to the previous layer. The attachment area is mostly sized on 3x3 or 5x5 cells. Cells input layer or previous layers of C-levels are linked to cells in the S-layer and V-layer. In addition, the each cell is linked to the S-cell layer. Finally, the S-cells are associated with C-cells each network link has a certain weight

VI. Testing Neocognitron Network

A. Basics

Neocognitron is a type of multi-layered hierarchical neural network that is used for recognition of handwritten characters. The network designed by Japanese Professor Kunihiko Fukushima in 1979. During the long period since its publication, the network has undergone many modifications, and today there are several different versions. The original version was without a teacher, but later versions were created with the teacher for whom it is necessary to create a special set of learning characters.

B. Network structure

As mentioned, this is a hierarchical network. This hierarchy is is based on that that the lower levels of the network detects the simplest symptoms. With each level of these symptoms are more complex. Each level always contain three layers: the S-layer, C-layer V-layer. The only exception is the zero level with only the input layer, which stores the input information. The individual layers consist of a number of areas that are

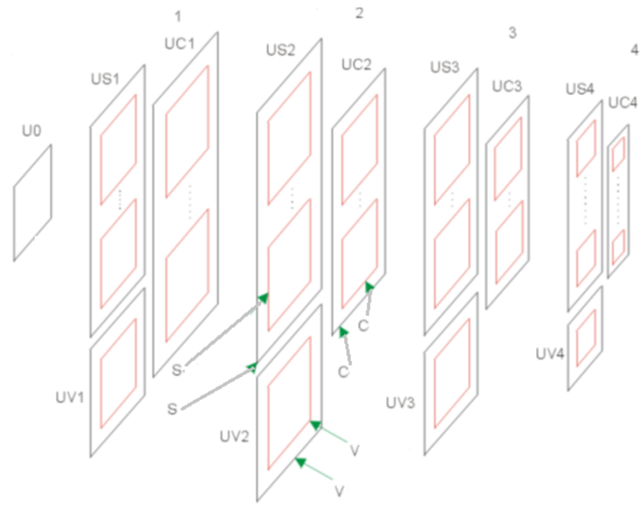

Fig.1. Neocognitron network structure

under the layers of either S, C or V. The area consists of two-dimensional array of cells. Network Neocognitron contains four basic types of cells: S-cells, C cells, the cells and receptor cells. The basic elements, ie individual cells, are already working with real non-negative values.

All areas contained in one layer have the same large array of cells. Likewise, they are dimensionally identical to the V-and S-surface area the same level. Example of areas and the cell layers for each level is shown in the following table (Table 1). This particular example is the classic application of neural networks Neocognitron the recognition of handwritten characters, namely the numbers 0 to 9 and capital letters A to Z. [1]

Layer	Cell size	Areas number	Cell number
U0	19 x 19	-	361
US1	19 x 19	12	4332
UV1	19 x 19	1	361
UC1	21 x 21	8	3528
US2	21 x 21	80	35 280
UV2	21 x 21	1	441
UC2	13 x 13	33	5 577
US3	13 x 13	97	16 393
UV3	13 x 13	1	169
UC3	7 x 7	64	3 136
US4	3 x 3	46	414
UV4	3 x 3	1	9
UC4	1 x 1	35	35
Total	-	380	70 045

Tab 1. Number of areas and cell of the neocognitron Network

C. Network connection

Neocognitron networks are characterized by a high density of connections between individual cells. Each cell is associated with a group of cells, the connecting areas of the previous layer. This connecting region is mostly size 3x3 or 5x5 cells. Cells input layer or layers of the previous C-levels are linked to cells in the S-layer V-layer. In addition,

D. Cells

As already mentioned, the neural networks Neocognitron has four types of cells. The first and simplest cells are receptor cells located in the input layer. Their function is to store information about one pixel input image. The number of receptor cells here gives us the maximum possible resolution of the input pattern

1) S-cells

Another type of cells are S-cells. Their task is to detect signs of pre-defined positions in the layer. Excitatory information obtained with each of its cell-attachment surface area of the previous C-level and also obtains information from the inhibitory V-cells. This information indicates the average activity in the area.

Formally, the output value of the S-cell is calculated according to equation. Formally, the output value of the S-cell is calculated according to equation.

$$u_{Sl}(n,k) = r_l(k) \cdot \varphi \left[\frac{1 + \sum_{\kappa=1}^{K_{Cl-1}} \sum_{\nu \in Al} a_l(\nu,\kappa,k) \cdot u_{Cl-1}(n+\nu,k)}{1 + \frac{r_l(k)}{1+r_l(k)} \cdot b_l(k) \cdot u_{Vl}(n)} \right]$$

l is the level number,

n coordinates of the cell,

a number of areas within the layers,

ν coordinates of cells within the connection area,

κ_Cl number of C-surfaces in C-layer

A_l region connecting the S-cells,

r_l selectivity,

φ threshold transfer function,

a_l and-weight,

b_l b-weight,

u_Vl output value of the V-cells.

An important parameter is primarily r_l, or selectivity. Its size affects the significance of linear inhibitory input component, which affects the ability to distinguish the deformed patterns. At high value of selectivity increases recognition accuracy, but reduces the ability to respond to more different patterns. On the contrary, reducing the selectivity roztřídíme and deformed patterns, but reduces the accuracy of sorting. Each S-layer may have a specific value of selectivity. Proper setting of this value is necessary for optimal functioning of neural networks.

2) V-cells

The main task is to transmit information about the average activity of connecting the C-space, the S-cell. The output value of the V-cell is described by the relation

$$u_{Vl}(n) = \sqrt{\sum_{\kappa=1}^{K_{Cl-1}} \sum_{\nu \in Al} c_l(\nu) \cdot u_{Cl-1}^2(n+\nu,\kappa)}$$

C – weight

3) C-cells

The task of the C-cells is to provide resistance to rotation and shift patterns. The activity of cells is directly proportional to the d-weight and value of the S-cells in the connection area. Connection areas of the individual C-cells in the S-layer overlap, so an S-cell affects more C-cells. This is guaranteed by a certain resistance as a result of the C-space is the blurring of the S-pattern area. Formally, the output value of C-cells described by the relation

$$u_{Cl}(n,k) = \psi \left[\sum_{\kappa \in K_{Sl}} \sum_{\nu \in Dl} d_l(\nu) \cdot u_{Sl}(n+\nu,\kappa) \right]$$

d_l the d-weight,

K_Sl number of S-surfaces in S-layer

D_l region connecting the C-cells

ψ transfer function.

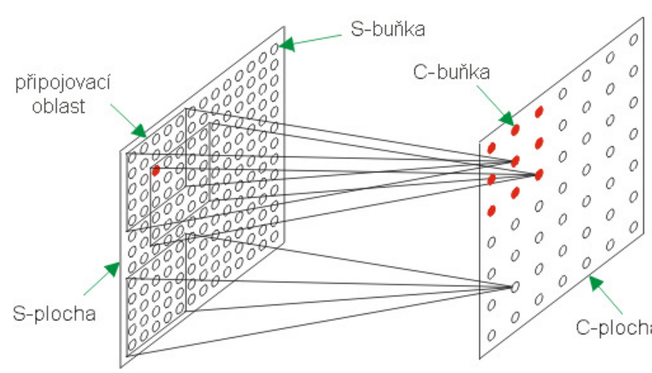

Fig 2. Overlap between C and S cells

E. Learning Neocognitron

Learning consists of presenting the training patterns and setting the modified weights (weights-and b-weights) so that the network is able to successfully detect the patterns. For learning neocognitron it is necessary to develop training models, which should directly correspond to recognized characters.

At the beginning of learning are all weights set to zero. Learning takes place from the lowest layers. First teaches the individual S-layer surface with the first level. The selected S-area select a single cell called a seed (seed) and the input is inserted into the desired pattern. For this seed is determined the seed weight. Since the weights in the same area with-shared, it automatically sets the seeds according to the scales with other cell-surface. Similarly, it is continued for another S-surfaces. Once all areas are taught, the learning process continues to the S-layer higher level.

For each S-plane is usually a training pattern, but there may be more. For each level of training pattern contains only a symptom characteristic of a given model, the network will recognize in this step. Symptoms should be set to detect a given level, what separates the characters. [5]

The Matlab algorithm was developed that perform convolutional network. Again, it was possible to choose the number of hidden layers, the number of cells in different layers, types of transfer functions and many other parameters. In this case of testing the MNIST database and own created database was used. Example learning curve when using a database MNIST: RMSE parameter represents mean square error, parameter CR success of process classification, which in the course of learning is changing (increasing). It can therefore be seen if the network during learning is improving or are no longer learning. For time reasons, the test sample was relatively small, about 500 patterns. Because of this the resulting success of the learned network classification can vary, so after each learning of the network the testing were made. Result of this is before mentioned percentage.

F. Neocognitron recognition

Recognition of the process itself is presented recognition patterns. The result is to determine the category to which the pattern belongs. So, put the pattern on the receptor cells of the input layer. Next, determine the value of the V-cell-layer in the first level. Values of S-cells can be then calculated, which detect the simplest symptoms. Next step is to process C-layer that blurs the image and then again analogically repeat process at a higher layer. The output of C-cell layer is the highest degree of similarity to the model presented with the category, has given C-cell represents. [3]

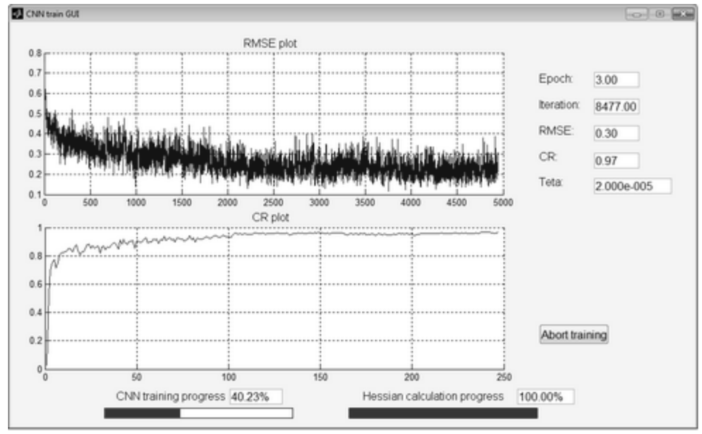

Fig. 1: Example of a neural network (97% success rate in identifying the test sample) in the final testing, the network showed a 97.4% success rate

The following table summarizes the results of learning and testing network database MNIST.

	Success rate at the end of learning	Success rate in the final test	Learning duration
500 samples	86 %	85,6 %	0,4 hour
5000 samples	91 %	91,9 %	2 hours
50000 samples	97 %	97,4 %	10 hours

Tab. 1: convolutional network (100 neurons), 500 = 500 patterns patterns from one classification group

If we assume a logarithmic dependence on number of patterns, so it could be calculated that for 100% success rate would be needed in this case about 19 000 patterns from each class. This number, however, in terms of practical implementation is inappropriate, regardless of the learning curve required to learn the network. This calculation is only indicative because of the small number of measurements. Another factor that has influence on the successful classification is the quality of the training set. If we classify numbers for example, a very common problem is the correct resolution number "7" and "2". In many cases, the problem with the classification has the human brain too. It is obvious that the success rate about 100% is unrealistic.

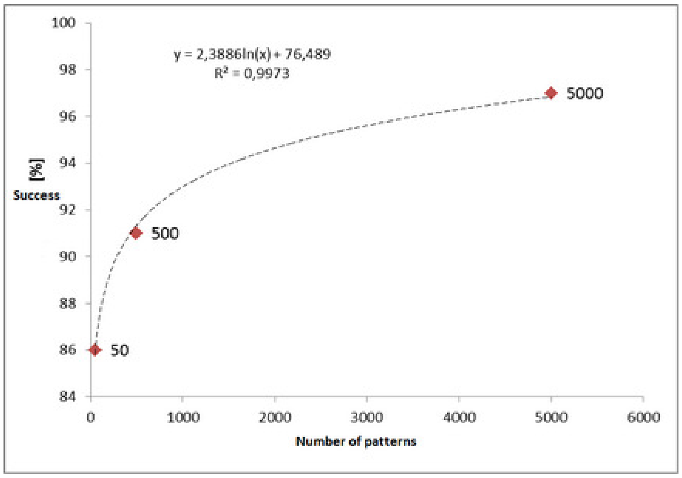

Fig. 2: The relationship between the success of classification and the number of patterns to learn the network

VII. Edge Detection

Edge detection in image is discipline for searching groups of pixels where brightness is changing considerably. This group of pixels is perceived as an edge of an object. Typical example are edges of form table. This detection in the OCR field is simplified because common forms used for datamining are originally in black and white color depth. Transform into the black and white color depth is one of the parts of prepare phase of the OCR process.

Transition between these two colors (the edge) can be described by transition function obtained by moving the picture. These transitions are looked up by edge detector algorithms [8].

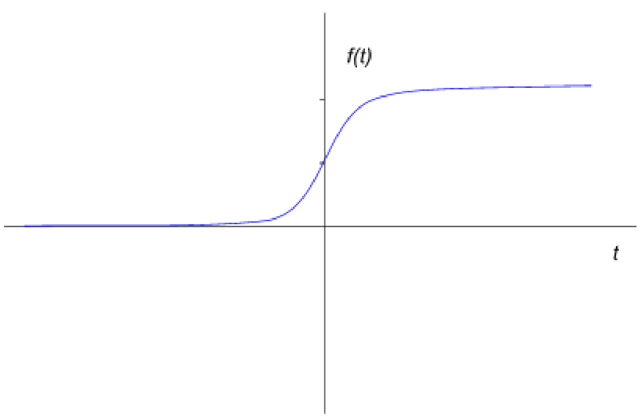

Fig. 3: The relationship between the success of classification and the number of patterns to learn the network

A. Edge detectors

There are many edge detectors. These detectors differ in many parameters and possible settings. Basic parameters are: way of edge search, edge representation,
accuration recognition, noise resistency and other image defects.

Edge detectors are divided according to the way of deciding about edges to category using first derivation of brightness function and detectors using second derivation of brightness function. In the first case the calculated edge gradient is compared with threshold and the result is determines the edge existence. Algorithms depending on second derivation of brightness function are comparing polarity change and its importance.

The most significant change in brightness function indicates the edge presence and is most strong in perpendicular direction. This is not used in praxis. The calculation is made only two or four directions. The calculation of gradient is implemented in practice as a convolution filter image. Each edge filters differs in core of filters. In the matrices are written values that determine which points are used to calculate the size and weight in this calculation. Size of matrices influents the resistance to noise and other image defects. Some of the most popular detectors are presented below.

Methods based on the second derivative of the brightness function tries to find passages of zero by this derivation. They use the fact that it is easier to find a passage through zero, rather than function extreme. Unfortunately, the second derivative is more sensitive to noise than the first, so it is appropriate to combine the calculation with such smoothing, which removes the maximum amount of noise and it does not damage the edges. These methods often work with the Laplace operator - Laplacian, which approximates the second omnidirectional derivative of the brightness function. The calculation therefore uses only one matrix. Its disadvantage is that it has a double response to the thin edge corresponding to the lines in the image.

A. Canny edge detector

Canny edge detector is known as the optimal edge detector. It is designed to meet three basic requirements for the edge detector.

• minimum number of errors – it has to find all edges in the image and the response in places that are not edges must be zero

• accuracy - the most accurate determining of the position of detected edge

• uniqueness - each edge has to be detected only once, here are not allowed duplicate edges

	G_x	G_y
Robertsův	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Prewittové	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobelův	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Robinsonův	$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
Kirsehův	$\begin{bmatrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{bmatrix}$	$\begin{bmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 2 \end{bmatrix}$
Frei-Chenův	$\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$
Prewittové(5x5)	$\begin{bmatrix} 2 & 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 & -2 \end{bmatrix}$	$\begin{bmatrix} -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}$

Fig. 4: The most common edge detectors

Edge detection procedure itself consists of several steps. It is necessary firstly eliminate noise – on the image is applied Gaussian noise, and is applied filter using a convolution mask. Second step is an application of Sobel operator. On the image is convolution mask of the Sobel detector applied, and the direction and edge gradient is found in this step. In the thinning process the detected edges are attenuated and placed to the right place, the principle of this phase is such that as the edge point is marked only such point, the neighboring points around the direction perpendicular to the gradient (direction of the gradient is known - it returns Sobel detector) have a lower value of the gradient. In the last step is provided image processing, its importance is to identify and assess the significance of found edges, the edges have a noise coming from a smaller gradient of the right edge, Cranny detector uses two thresholds, if the value exceeds the higher threshold gradient edge is recognized. If the value is lower than lower threshold gradient the edge is not recognized. If the value is between thresholds the edge is recognized only when is adjacent with other edge.

Cranny edge detector is currently regarded as the most optimal detector.

B. *Hough transform*

Hough transform is a method that is helping to find a parametric description of objects in the image. The disadvantage is that it is needed to know in advance analytical description of the shape object that is looked for. Therefore, this method is used to find familiar objects such as lines, circles, ellipses, and these composite figures as squares, triangles and so on. The biggest advantage of this method is its robustness and resistance to noise or disturbed data. The method uses a conversion from Cartesian to polar coordinate system. Detection of lines such as the document is used to detect rotation of the document and the document can use it properly and turn the handle.

$$r = x \cdot cos\theta + y \cdot \theta$$

Fig. 5 example of parametric description

r is the length from the normal line to the origin and θ is the angle between the normal and the axis x.

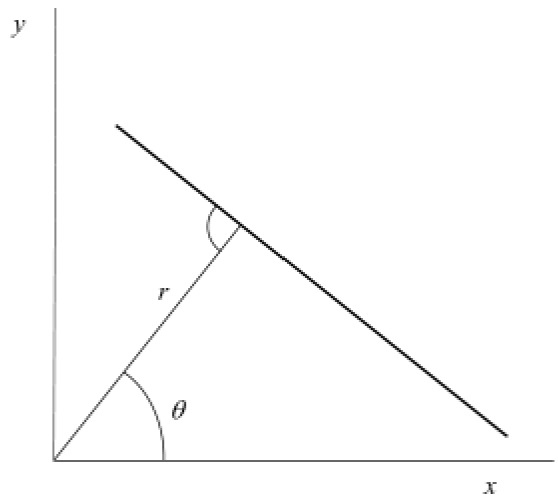

Fig 6. Parametric description of the line

The actual transformation is performed so that the parameters x and y each entry point image that represents a point or line segment, are substituted into the equation (1.1). The value of θ is gradually substituting each possible value - interval <0, 360> (mathematically it is an infinite set of values of real numbers representing the angle θ, but in case of implementation are substituting values according to the sensitivity settings Hough transform) and the value of r is interpolating. So in the parametric space curves are generated for each entry point one. If entry points are collinear, the parametric space curves intersect at point representing the most common search parameters line.

C. *Representation of edges*

Representation of edges gives us the apparatus with which is possible to store and present data provided by edge detectors. Presentation of these edges is in most cases inadequate and it is needed to continue in processing these edges and save afterwards.

If we manage to get the contours of the image, it is needed to process these contours. One of the biggest criteria for the processing of these contours is their nature. If it is needed to store data, where the assumption of long lines or a small number of sharp bends is beneficial to use method of storing by mathematical functions. If it is necessary to process raster data where we can expect a large number of sharp changes in direction, the load noise and image defects or very irregular shapes, it is better to use string codes.

D. String codes

The principle of the code string is that the path from one point to neighboring point of described contours can be identified - for example number. If considered operating at the level of individual pixels, it is necessary to select according to consistency mark 4 to 8 directions. Freeman chain code works with 8 lines by marking them with numbers from 0 to 7 These trends can be seen as opportunities to turn a lady on the board. Then the chain code contours starting at S will look like this {4,4,3,2,3,6,6,7,4,6,0,0,0,1}

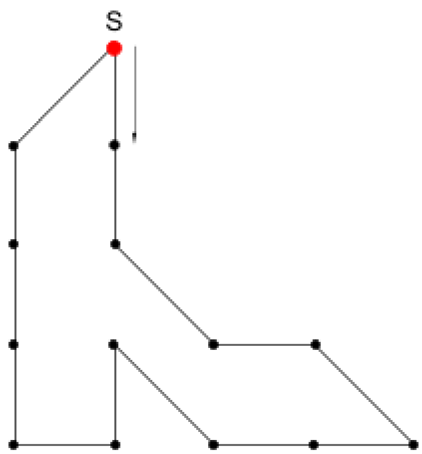

Fig 7. Contours of chain code

VIII. CONCLUSION

Tests and their results suggest that an important factor for correct classification using neural networks is not only the parameters of neural networks, but also sufficiently large training set and a chosen approach to the image processing of submitted designs. Best results in terms of successful classification were achieved by using a convolutional neural network, since it is resistant to translation, rotation or scaling the input pattern. Perceptron network was significantly more sensitive to these changes. For increasing the percentage success was given appropriate preprocessing, which, however, significantly increased required time.

In the first part of this article is described background knowledge necessary for understanding problematic with basic principles. There is covered neural networks which were used in tests in detail. In this part is also mentioned their advantages and disadvantages.

Tests were made on two databases (MNIST and custom database of samples, approximately 60 thousand of patterns). The greatest attention was paid to the recognition of handwritten digits, when after appropriate learning, the network can correctly classify over 97% of unknown patterns (database MNIST). The same network configuration was then applied to learn from own database (about 264 patterns), which achieved greatest success rate was 84%.

In the next part, the networks were tested on the capitals of the Czech alphabet (including diacritical marks) and in combination with numbers, the results ranged between 70% and 80%. Depending on number of classification groups, the percentage success is different.

From tests is obvious that more classification groups leads to less accurate results. Reason of this was in very small training set. So we tried to expand the existing database by applying shift and rotation. So we got the original 264 model three times, the success rate has improved only a few percent units, apparently because of resistance to this type of transformation.

So we would suggest a more extensive database of models and more experimental tests of the network, setting appropriate configuration and parameters. Achieved level of classification can be improved by deepening of knowledge of the context of the final text, or using a database of names, addresses, etc

ACKNOWLEDGMENT

This research was supported by the Internal grant agency (IGA/29/FAI/11/D)

REFERENCES

[1] I, Zelinka. *Artificial Intelligence:* Volume 1. Brno : Zlin: Vutium, 1998
[2] V. Cappelini, *Time Varying Image Processing*, Elsevier Science, 1997.
[3] S.Haykin, *Neural Networks and Learning Machines* . Third Edition - New York : Prentice Hall 2009
[5] A.K. Jain, *Fundamentals of Digital image processing*, Prentice Hall, 1988.
[6] R.C. Gonzales, R.E. Woods, *Digital Image processing*, Prentice Hall, 2002.
[7] B.E.Boser, I.M.Guyton, V.N.Vapnik, *A training algorithm for optimal margin classifiers* New York 1992
[8] B. Javidi, *Image Recognition and Classification*, CRC Press, 2002.
[9] D. Sun, *Computer Vision Technology for Food Quality Evaluation*, Academic Press, 2007
[10] R.Gonzales, C; WOODS, R E; Eddins *Digital image processing using MATLAB*. Upper Saddle River : Pearson/Prentice Hall, 2004. 609 s.
[11] NAGEL, Christian, et al. *C Sharp 2008 Programming proffesionally*.. Brno : Computer press, 2009. 1904 s. ISBN 978-80-251-2401-7.
[12] LECUN, Yann, et al. *Gradient-Based Learning Applied to Document Recognition*. Proceedings of the IEEE. 1998
[13] RŮŽEK, Václav. *Algorithms for hand written text recognition*. Zlin
[14] BERÁNEK, Jan. *Methods of detection and representation of edges in image*. Brno, 2007.
[15] BENEŠ, Miroslav. *Searching in text* [online]. WWW:
[16] J.Csirik, P. Bertholet, H.Bunke *Pattern Recognition in Wireless Sensor Networks in Presence of Sensor*
[17] C.M.Bishop, *Neural networks for pattern recognition,* Oxford university press 1995
[18] C.Cernazanu-Glavan, S. Holban, *Determining the Optimal Percent of Negative Examples used in Training the Multilayer Perceptron Neural Network,* Proceedings of the 10th WSEAS International Conference on NEURAL NETWORKS (NN'09), Prague
[19] S. Iranmanesh, *A Differential Adaptive Learning Rate Method for Back-Propagation Neural Network,* Proceedings of the 10th WSEAS International Conference on NEURAL NETWORKS (NN'09), Prague
[20] Aleksandar Miloksavljevic, *An algorithm for automatic generation of fuzzy neural network based on perception frame* Proceedings of the 9th WSEAS International Conference on NEURAL NETWORKS (NN'08), Bulgaria
[21] C. Botoca, G.Budura, *Symbol decision equalizer using a radial basis 7th WSEAS International Conference on NEURAL NETWORKS (NN'06)*