Non-Conventional Approaches to Feature Extraction for Face Recognition

Jozef Ban[1], Matej Féder[2], Miloš Oravec[2], Jarmila Pavlovičová[1]

[1] Department of Telecommunications, [2] Department of Applied Informatics and Information Technology, Faculty of Electrical Engineering and Information Technology, STU in Bratislava, Slovak Republic

jozef.ban@stuba.sk, matej.feder@stuba.sk, milos.oravec@stuba.sk, jarmila.pavlovicova@stuba.sk

Abstract: This paper deals with human face recognition based on the use of neural networks, such as MLP (multi-layer perceptron), RBF (radial basis function) network, and SVM (support vector machine) methods. The methods are tested on the MIT (Massachusetts Institute of Technology) face database. We use non-conventional methods of feature extraction for the MIT face images in order to improve the recognition results. These methods use so called HLO and INDEX images. HLO images are generated by feature extraction of the MLP neural network in auto-association mode, and INDEX images are formed by a self-organized map used for image vector quantization. We propose novel methods based on HLO and INDEX images with SVM classifier. We also analyze the impact of adding noise to the learning process.

Keywords: human face recognition; feature extraction; multilayer perceptron; RBF networks; support vector machines

1 Introduction

Biometric face recognition [1] is a difficult and developing task even for the most advanced computer technology. While humans can recognize a familiar face in various lights and perspectives, there remain barriers to effective recognition by computers.

In this paper, we use and compare three methods of biometric face recognition. The first method is a neural network - multilayer perceptron (MLP), the second one is a radial basis function (RBF) neural network, and the third one is a support vector machine (SVM). For input data we used images of 64x60 pixels taken from the MIT (the Massachusetts Institute of Technology) database. The images were divided into two sets. The first set was designed for training, and the second (more extensive) set was designed for the recognition test itself. We attempted to achieve the best results of biometric face recognition by setting the appropriate parameters for the MLP, RBF and SVM methods. For the MLP and RBF methods we used Matlab7 (http://www.mathworks.com/) software (Neural Network Toolbox). For the SVM method we used the Libsvm (A Library for Support Vector machines) freeware (http://www.csie.ntu.edu.tw/~cjlin/libsvm/).

The aim of the work is an evaluation of achieved results on the unmodified images taken from the MIT database and the analysis of the impact of proposed non-conventional methods of image feature extraction through HLO and INDEX images. Appropriate feature extraction [2], [3] reduces storage requirements and improves the computational and time complexity of used methods. We also evaluated the impact of adding Gaussian white noise to the training set of images, and its effect on the recognition results.

2 Neural Networks

A neural network [4], [5], [6], [7] is a massive parallel processor generally used to store experimental information for later use. It simulates the human brain in two aspects:

- a neural network obtains the information from an environment by a learning process,
- connections among neurons (synaptic weights SW) are used for information storage.

2.1 Multilayer Perceptron

The multilayer perceptron (MLP) [5], [6] is the first method used for face recognition in this paper.

This type of network consists of an input layer of elements which distribute input signals to the network, one or more hidden layers and one output layer of computational elements.

2.1.1 Multilayer Perceptron Characteristics

The existence of one (Fig. 1) or more hidden layers allows the network to learn complicated tasks, because it selects the most important features from the input samples. The term “hidden neuron” refers to a neuron that is not a part of either the input or output layer, thus inaccessible from the outside world. An important feature of this method is a high degree of network interconnection determined by network synapses. Another significant characteristic is that the model of each neuron in the network contains nonlinearity in its output. This nonlinearity must be smooth, i.e. differentiable everywhere.

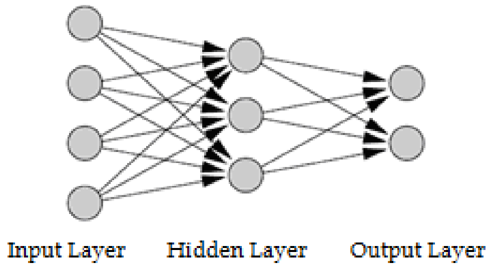

Figure 1
MLP with one hidden layer

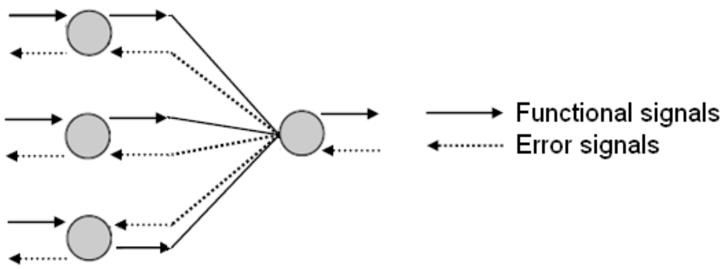

Figure 2
Functional and error signals in MLP

In the given part of the multilayer perceptron (Fig. 2), it is possible to notice the functional and error signals. The functional signal propagates forward starting at the network input and ending at the network output as an output signal. The error signal originates in the output neurons during learning and propagates backward.

Hidden and output neurons perform two computations during their training. The first is the computation of a functional signal which is the output of the neuron. It is expressed as a nonlinear function of the input signals and synaptic weights connected to this neuron. The second computation is an estimation of the instantaneous gradient vector, i.e. the error surface gradient with regard to the weights which are connected to the given neuron. This vector is necessary for the back propagation phase of errors.

2.1.2 Backpropagation Algorithm

The MLP is trained in a supervised manner (by a teacher). It uses a back propagation algorithm [5], [6]. The error signal at the output of neuron j for n-th training sample (iteration) is defined as:

$$e_j(n) = d_j(n) - y_j(n) \tag{1}$$

supposing that neuron j is an output neuron, d_j(n) is a desired and y_j(n) actual response.

The internal activity of neuron *j* is:

$$v_j(n) = \sum_{i=0}^{p} w_{ji}(n) y_i(n) \tag{2}$$

where p is the number of inputs of the neuron j and w_{ji}(n) is a synaptic weight connecting the output of neuron i to the input of neuron j in iteration n. The functional signal at the output of the neuron j with the activation function $\varphi_j()$ for iteration n is:

$$y_j(n) = \varphi_j(v_j(n)) \tag{3}$$

The adjustment Δw_{ji}(n) of the weight connecting neurons i and j is defined by delta rule:

$$\begin{pmatrix} \text{adjustment} \\ \text{of weight} \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} \text{parameter} \\ \text{of fast learning} \\ \eta \end{pmatrix} * \begin{pmatrix} \text{local} \\ \text{gradient} \\ \delta_\mathrm{j}(n) \end{pmatrix} * \begin{pmatrix} \text{input signal} \\ \text{of neuron j} \\ y_\mathrm{i}(n) \end{pmatrix} \tag{4}$$

Thus, the backpropagation algorithm adjusts weight w_{ji}(n) by value Δw_{ji}(n), which is proportional to the instantaneous gradient $d\varepsilon$(n)/dw_{ji}(n).

2.1.3 Stopping Criterion for Learning

For the backpropagation algorithm, several stopping criteria exist. It is possible to set a maximum number of training cycles or some kind of a maximal output error. We used a cross-validation as the stopping criterion. It consists of dividing the available data to a training and a test sets. After a period of training, the network is tested on a test set in order to examine its generalization properties. The process of learning stops when there is an increase of error for the test set (this method is also known as the early stopping method of training). At that time the network has reached its maximum ability to generalize.

2.2 Radial Basis Function Network

A radial basis function network (RBF network) is a neural network in which the hidden neurons represent a set of functions forming a base for input vector transformation into the hidden neuron space. These functions are called RBF – radial basis functions [6], [8], [9], [10], [11]. The learning process is equivalent to searching the space which best approximates the training data. Generalization is then equivalent to a utilization of this space for an interpolation of testing data.

In many cases, better results are achieved by RBF networks than by networks with sigmoidal activation functions (e.g. MLP – multilayer perceptron). One possible reason is that RBF activation functions respond better to the receptive fields of real neurons [6], [8].

An example of an RBF network is shown in Fig. 3. Formally, an RBF network can be described as follows [12]:

$$f(x) = w_0 + \sum_{i=1}^{m} w_i h_i(\mathbf{x}) \tag{5}$$

where $\mathbf{x}$ is a parameter of RB activation function h_i and w_i are weights. The output of the network is a linear combination of RBFs.

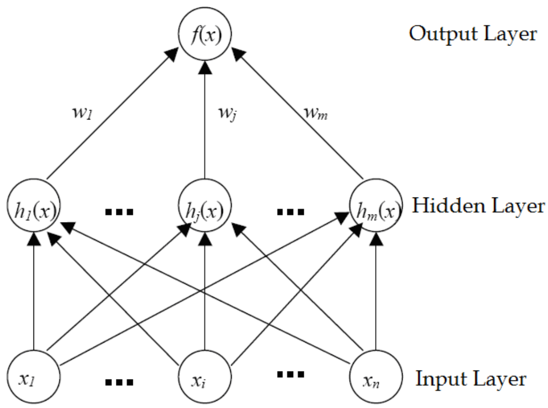

Figure 3
RBF network [12]

The training set for an RBF network consists of N input-output pairs $(\mathbf{x}_k,\boldsymbol{d}_k)$, $k=1,\dots,N$, where $\mathbf{x}_k \in R^p$ is input and $\boldsymbol{d}_k \in R^q$ is a desired response.

The RBF net is trained in three steps [6]:

- The first step is a determination of the hidden elements centers, which are represented by the weights between input and hidden layer. An easy solution is a random selection of n_0 points of the inputs, which we set as the centre values. Another approach generates uniform distribution of the centers in the input space.

  However, it is desirable that the centers comply with the structure of the input data – this requires the use of more complex algorithms in order to set up the centers. This task falls within a category of self-organizing learning.

- The second (optional) step is the setup of additional parameters of RBFs. Let us mention the often-used (even in our experiments) Gaussian radial basis function:

$$\phi(x) = \exp\left(-\frac{\|x-c\|^2}{\sigma^2}\right) \tag{6}$$

Its parameter σ represents a width of function ϕ which determines a radial space around the centre c in which a hidden element has a rational response. Widths of the functions influence the generalizing capacity of the net – the smaller width, the worse generalization is to be expected. Parameter σ was the object of our tests.

- In the third step of learning we set up values w, through the minimization of the error function E:

$$E(\mathbf{W}) = \frac{1}{2}\sum_{l=1}^{N}\|\mathbf{d}_l - f(\mathbf{x}_1)\|^2 = \frac{1}{2}\sum_{l=1}^{N}\sum_{k=1}^{q}(d_{kl} - f_k(\mathbf{x}_l))^2 \tag{7}$$

where $f_k(\mathbf{x}_l) = \mathbf{y}_l$ is a response (output) of the network to input $\mathbf{x}_l$, $f_k(\mathbf{x}_l) = y_{kl}$ is a response (output) of k-th output element to input $\mathbf{x}_l$, $\mathbf{W}$ is matrix $[w_{sr}]$ – i.e. with units w_{sr}, $s = 1,...,q$ and $r = 1,...,n_0$.

A solution obtained by $\frac{\partial E_3}{\partial w_{sr}} = 0$ is as follows:

$$\mathrm{w}_{\mathrm{sr}} = \sum_{j=1}^{n_0}\left[\Phi^+\right]_{jr}\left(\sum_{l=1}^{N}\phi_j(\mathbf{x}_l)d_{kl}\right) \tag{8}$$

where matrix $\mathbf{\Phi}$ is defined as

$$[\Phi]_{jr} = \sum_{l=1}^{N}\phi_j(\mathbf{x}_l)\phi_r(\mathbf{x}_l) \tag{9}$$

and $\mathbf{\Phi}^+$ is a pseudoinverse matrix.

3 Support Vector Machine

Support vector machines (SVM) [13], [14], [15], [16], [17] are based on the concept of decision planes that define optimal boundaries. The optimal boundary separates two different sets (sets of objects having different class membership) and is located as to achieve the largest distance between these sets. In the case of linearly nonseparable sets, a SVM uses kernel methods.

3.1 Kernel Methods

The soft margin method is an extension of a SVM within linear methods [18]. The kernel method is a method of finding non-linear thresholds. The basic concept of the kernel method is the transformation of vector space into a high-dimensional space. Let us consider the linearly non-separable example shown in Fig. 4(a) [13]. If the two-dimensional space is transformed into a three-dimensional space (Fig. 4(b)), the black and white vectors become linearly separable.

Φ is the transformation into a multidimensional space. The space which is to be transformed should match the distance defined in the transformed space and is related to the distance in the original space. Kernel function $K(\mathbf{x},\mathbf{x'})$, which meets both conditions is defined as follows:

$$K(\mathbf{x},\mathbf{x'}) = \Phi(\mathbf{x})^T \Phi(\mathbf{x'}) \tag{10}$$

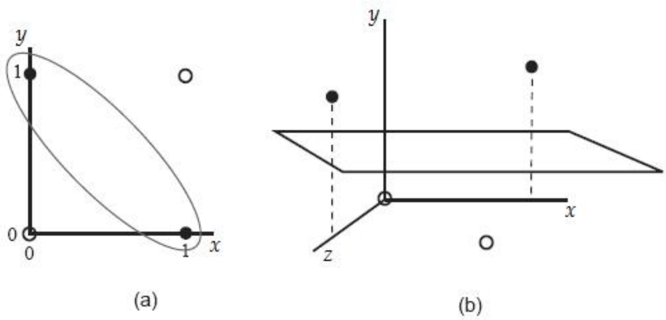

Figure 4

Linearly non-separable vector space (a), linearly separable vector space (b) [13]

This equation indicates that kernel function is equivalent to the distance between ***x*** and ***x'*** measured in high-dimensional space, transformed by Φ . If we measure the margin by the kernel function and perform an optimization, a nonlinear boundary is obtained.

The transformed space's threshold can be found by

$$\mathbf{w}^{\mathbf{T}}\Phi(\mathbf{x}) + b = 0 \tag{11}$$

It can be then formulated as:

$$\sum_i \alpha_i y_i \Phi(\mathbf{x}_\mathbf{i}^\mathbf{T})\Phi(\mathbf{x}) + b = \sum_i \alpha_i y_i K(\mathbf{x}_\mathbf{i},\mathbf{x}) + b = 0 \tag{12}$$

Optimization function in transformed space is also obtained by substitution of $\mathbf{x}_i^T\mathbf{x}_j$ to $K(\mathbf{x}_i,\mathbf{x}_j)$. These results prove the fact that all the calculations can be obtained just by using $K(\mathbf{x}_i, \mathbf{x}_j)$, without the exact formulation of Φ and the transformed space. K has to be positive definite (the sufficient condition).

Several examples of kernel function have been known, e.g.:

Polynomial kernel $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^{\mathrm{T}}\mathbf{x}' + 1)^p$ (13)

Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\sigma^2}\right)$ (14)

4 Used Data and Methods

4.1 The MIT Face Database

For the simulation of the biometric methods of face recognition we used a face database developed at MIT (the Massachusetts Institute of Technology) (http://web.mit.edu/). The MIT database contains 432 images. It consists of 16 subjects (Fig. 5), while each subject is represented by 27 images (Fig. 6). Each subject is distinguished by various head tilts, illumination and distances from the lens of the camera. Tests of individual methods use 256 level grayscale images of dimensions 64x60 pixels.

Figure 5
Face database subjects

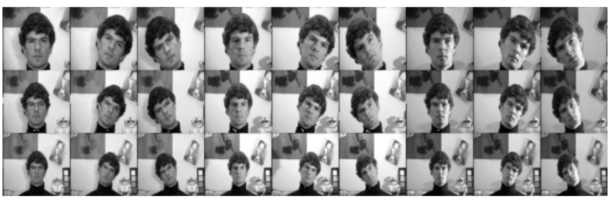

Figure 6
Various samples of the same subject

4.2 Novel Methods Based on Non-conventional Approaches to Feature Extraction

In order to improve recognition results, we use non-conventional methods of feature extraction for the MIT face images. These methods use so called HLO and INDEX images. HLO images are generated by feature extraction using the MLP neural network in auto-association mode and INDEX images are formed by a self-organized map (SOM) used for image vector quantization. We propose novel methods based on HLO and INDEX images with SVM classifier. The formation of HLO and INDEX images is described in sections 4.3 and 4.4. An overview of the methods used (two proposed methods using feature extraction and standard methods of direct classification) is shown in Fig. 7.

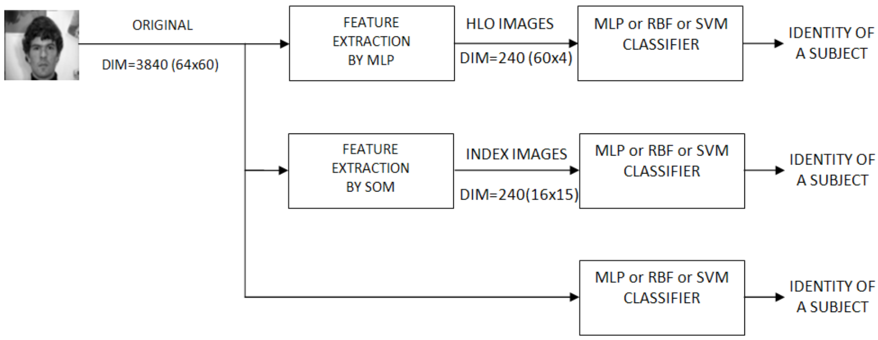

Figure 7

Overview of methods used for simulations:

- two proposed methods using feature extraction by MLP (top) and SOM (middle)
- standard methods of direct classification used for comparison purposes (bottom)

4.3 HLO Images

HLO (Hidden layer outputs) images [19] are the result of the feature extraction method by the MLP in auto-associative mode. Its configuration and utilization for feature extraction is shown in Fig. 8. Input face images of dimensions 64x60 pixels are divided into 16x15 blocks. The MLP configuration is 16x15-15-16x15 (i.e. 240 input and output neurons and 15 hidden neurons). For the MLP 240-15-240 and 64x60 images divided into blocks 16x15 pixels we obtain 16 blocks from each image and each block is represented by 15 hidden layer outputs. Each face image is thus represented by 240 hidden layer outputs, which are used for HLO image formation – resulting in 60x4 HLO images. This corresponds to a bit rate of 0.5 bit/pixel compared to 8 bit/pixel in the original images.

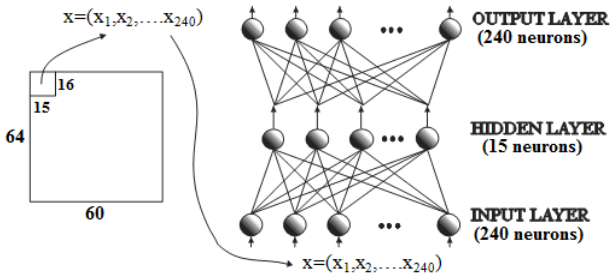

Figure 8
64x60 face image is divided into 16x15 blocks and then fed into the MLP network 240-15-240

Fig. 9 represents a sample of HLO images for 16 subjects of the MIT face database from Fig. 5.

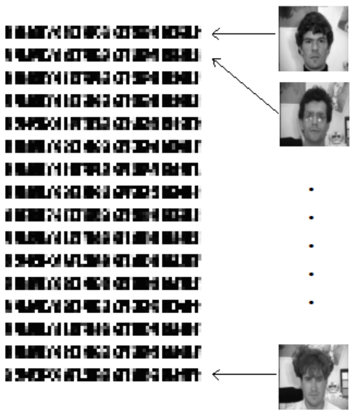

Figure 9
60x4 HLO images for 16 MIT face database subjects

4.4 INDEX Images

INDEX images result from feature extraction method based on the vector quantization (VQ) of images using Kohonen self-organizing map for codebook design. This method is described in [20]. Vector quantization is performed on 64x60 face images dividing the originals into 4x4 blocks. After vector quantization, all 240 indexes corresponding to all blocks of the face image are formed into a two-dimensional array of dimensions 16x15. For the image vector quantization, the configuration of the self-organizing map with 16x16 neurons with 16-dimensional weight vectors was used. It again corresponds to a bit rate of 0.5 bit/pixel. INDEX images corresponding to face images from Fig. 5 are shown in Fig. 10.

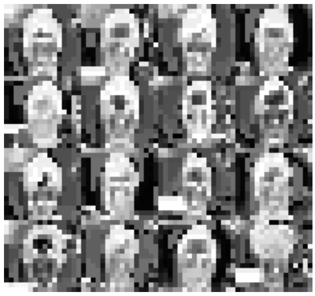

Figure 10
MIT database INDEX images (each of 16 images is 16x15 pixels)

4.5 Noise Modification of Images

We attempted to analyze the impact of adding noise to the learning process. Adding new training samples modified by the noise can lead to better convergence for some methods. For the modification of the MIT database images, Gaussian white noise [21] was used. It was applied with variance $v=0.02$ and zero mean $m=0$. Gaussian white noise for a pixel x is defined as follows:

$$p_q(x) = (2\pi)^{-1/2} e^{-(x-m)^2/2v^2} \tag{15}$$

An example of the original and the noisy image is shown in Fig. 11.

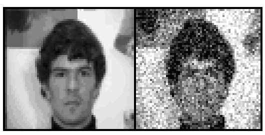

Figure 11
Original and noisy MIT image 64x60 pixels, Gaussian white noise, v=0.02

4.6 Training and Test MIT Sets

For our simulations, two main training sets *a11* and *a11+112+113* were created. These training sets are described in Fig. 12. The set *a11* consists of all 111, 211 and 311 images. The set *a11+112+113* consists of a11 set plus all 112 and 113 images. These two main sets were extended by images modified by noise (according to Section 4.5). The same holds also for HLO and INDEX training sets.

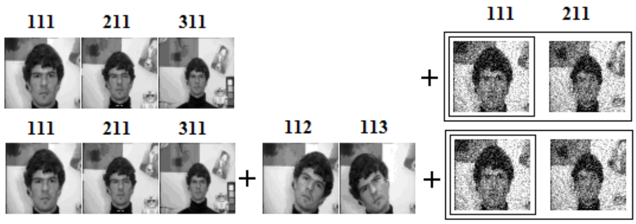

Figure 12
Illustration of the MIT database images division into training and test sets - a11 (111, 211, 311), number of images is 48 (3 views for each of the 16 subjects), a11+112+113 represented by 80 images (5 views for each of the 16 subjects)

4.7 Parameters of MLP, RBF and SVM Methods

Simulation results are influenced by the proper settings of the parameters of the individual methods:

- **MLP**

  The topology of the neural network, the learning rate η and training method (gradient descent back propagation, gradient descent with momentum and adaptive learning rate back propagation) are the parameters affecting the training process.

- **RBF**

  A decisive impact on the success of human face recognition for the RBF method is influenced by the number of neurons in the hidden layer of the network, and in our case also by a variance parameter of Gaussian curve σ .

- **SVM**

  The success of human face recognition in the SVM method depends on the setting of value *C* as a parameter of the SVM method, and on the optimization of the value γ (the parameter of the used kernel function).

5 Simulation Results

Three standard face recognition methods, MLP, RBF and SVM, used for the direct classification (the bottom method in Fig. 7) of the input images (i.e. without feature extraction), were compared. Figure 13 and Table 1 show the recognition results achieved by these methods. The SVM is the most successful method for face recognition (without feature extraction), with an average value of recognition efficiency of 78.32%. The comparable results were achieved also by a method using the RBF network with an average value of 70.03%. The MLP method in comparison to the SVM and RBF achieved unstable and insufficient results with an average value of 44.60%.

We also analyzed the impact of extending training sets by modified images (with Gaussian noise). This expansion helped in several simulations with the methods MLP, SVM, and significantly increased the success of recognition with the RBF method.

The essential goal of the paper was to propose the methods using feature extraction through HLO and INDEX images followed by SVM classifier (the top and the middle method in Fig. 7). Interesting results were achieved by simulations for the proposed approaches. In Figure 13, the simulations on a smaller training set (a11) are denoted by the color black. The value of 72.14% (black underlined) obtained on the 64x60 pixel images by the RBF method has been overcome in five cases using HLO and INDEX images (SVM classifier was used in four cases). In the case of the larger training set (a11+112+113, brown underlined), the best value of 92.33% has been overcome only once - with the use of HLO images. In this case, however, the best recognition efficiency was achieved (93.18%).

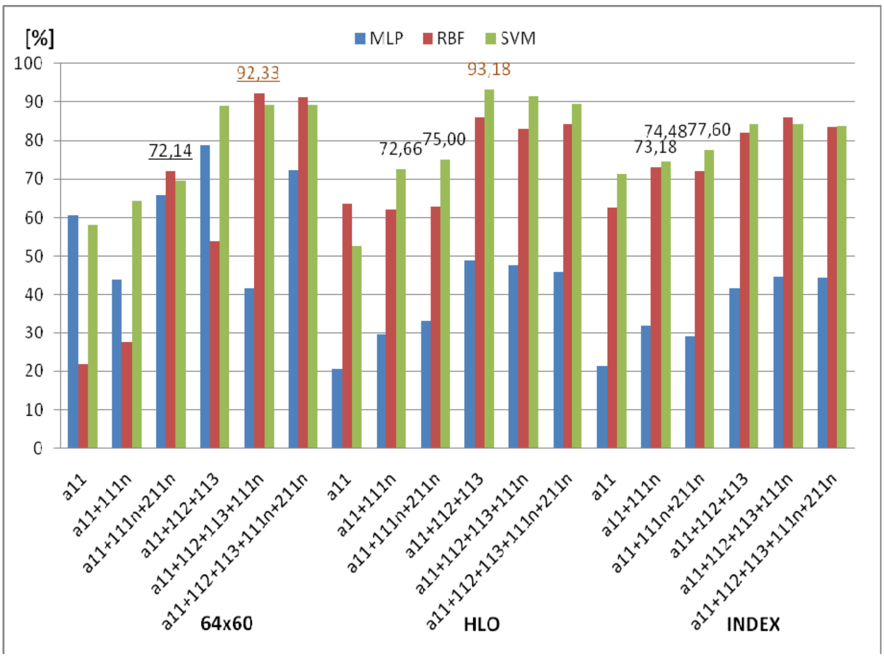

Figure 13

Graphic comparison of the results by methods MLP, RBF and SVM

Table I
The results of the MLP, RBF and SVM methods

		Recognition efficiency [%]		
Image	Training set	MLP	RBF	SVM
64x60	a11	60,6771	21,875	58,0729
64x60	a11+111n	44,03	27,6042	64,3229
64x60	a11+111n+211n	65,9	72,1354	69,5313
64x60	a11+112+113	78,6932	53,9773	88,9205
64x60	a11+112+113+111n	41,7	92,3295	89,2405
64x60	a11+112+113+111n+211n	72,4	91,1932	89,2405
HLO	a11	20,5729	63,5417	52,6042
HLO	a11+111n	29,4271	62,2396	72,6563
HLO	a11+111n+211n	33,3333	62,7604	75
HLO	a11+112+113	48,8636	86,0795	93,1818
HLO	a11+112+113+111n	47,7273	82,9545	91,4773
HLO	a11+112+113+111n+211n	46,0227	84,375	89,4886
INDEX	a11	21,3542	62,5	71,3542
INDEX	a11+111n	32,0313	73,1771	74,4792
INDEX	a11+111n+211n	28,9063	72,1354	77,6042
INDEX	a11+112+113	41,7614	82,1023	84,375
INDEX	a11+112+113+111n	44,7917	86,0795	84,375
INDEX	a11+112+113+111n+211n	44,5313	83,5227	83,8068

Conclusion

In this paper, we proposed novel methods for face recognition based on non-conventional feature extraction followed by the SVM classifier. We used MLP for the formation of HLO (hidden layer output) images and Kohonen SOM for the formation of INDEX images. Such representation was fed into MLP, RBF and SVM classifiers. The presented results show excellent recognition efficiency and they were compared to results of three standard methods using direct classification by MLP, RBF and SVM. For a small training set, results on the proposed methods using HLO and INDEX images overcame in five cases results of direct classification. For a larger training set, the method using HLO images achieved the best results of all. SVM was the most successful classifier for HLO and INDEX images. The advantages of the proposed methods are as follows: the reduction of storage requirements, the speed-up of the learning process and the improvement of computational and time complexity.

Acknowledgement

Research described in the paper was done within the grants No. 1/0214/10 and 1/0961/11 of the Slovak Grant Agency VEGA.

References

[1] Jain, A. K., Ross, A., Prabhakar, S.: An Introduction to Biometric Recognition, IEEE Trans. Circuits and Systems for Video Technology, Vol. 14, No. 1, Jan. 2004, pp. 4-20

[2] Puyati, W., Walairacht, S., Walairacht, A.: PCA in Wavelet Domain for Face Recognition, The 8th International Conference Advanced Communication Technology, p. 455, Phoenix Park, ISBN: 89-5519-129-4, 2006

[3] Prema, R., Thirunadanasikamani, K., Suguna, R.: A Novel Feature Extraction Scheme for Face Recognition, 2010 International Conference on Signal and Image Processing (ICSIP), pp. 502-505, 15-17 Dec. 2010

[4] Bishop, C. M.: Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1996

[5] Haykin, S.: Neural Networks - A Comprehensive Foundation, New York: Macmillan College Publishing Company, 1994

[6] Oravec, M., Polec, J., Marchevský, S.: Neural Networks for Digital Signal Processing (in Slovak), Bratislava, 1998

[7] Oravec, M., Petráš, M., Pilka, F.: Video Traffic Prediction Using Neural Networks, Acta Polytechnica Hungarica, Journal of Applied Sciences, Hungary, Vol. 5, Issue 4, pp. 59-78, 2008

[8] Hlaváčková, K., Neruda, R.: Radial Basis Function Networks, Neural Network World, No. 1, 1993, pp. 93-102

[9] Poggio, T., Girosi, F.: Networks for Approximation and Learning, Proc. of the IEEE, Vol. 78, No. 9, Sept. 1990, pp. 1481-1497

[10] Zhang, Y., Xue, Z. M.: RBF Neural Network Application to Face Recognition, 2010 International Conference on Challenges in Environmental Science and Computer Engineering (CESCE), Vol. 2, pp. 381-384, 6-7 March 2010

[11] Benyó, B., Somogyi, P., Paláncz, B.: Classification of Cerebral Blood Flow Oscillation, Acta Polytechnica Hungarica, Journal of Applied Sciences, Hungary, Vol. 3, Issue 1, pp. 159-174, 2006

[12] Mark, J. L. Orr: Introduction to Radial Basis Function Networks. Centre for Cognitive Science, University of Edinburgh, April 1996

[13] Asano, A.: Pattern Information Processing, (lecture 2006 Autumn semester), Hiroshima University, Japan, 2006, http://laskin.mis.hiroshima-u.ac.jp/Kougi/06a/PIP/PIP12pr.pdf

[14] Hsu, C. W., Chang, C. C. Lin, C. J.: A Practical Guide to Support Vector Classification. Department of Computer Science and Information Engineering, National Taiwan University, 2003

[15] Müller, K. R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An Introduction to Kernel-based Learning Algorithms. IEEE Transactions on Neural Networks, Vol. 12, No. 2, March 2001, pp. 181-201

[16] Boser, B. Guyon, I. Vapnik, V.: A Training Algorithm for Optimal Margin Classifiers, In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM Press, 1992, pp. 144-152

[17] Faruqe, M. O., Al Mehadi Hasan, M.: Face Recognition Using PCA and SVM, 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication, ASID 2009, pp. 97-101, 20-22 Aug. 2009

[18] Horváth, G.: Kernel CMAC: an Efficient Neural Network for Classification and Regression, Acta Polytechnica Hungarica, Journal of Applied Sciences, Hungary, Vol. 3, Issue 1, pp. 5-20, 2006

[19] Oravec, M., Pavlovičová, J.: Feature Extraction by Multilayer Perceptron - Visualization of Internal Representation of Input Data, The Seventh IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2007), August 29-31, 2007, Palma de Mallorca, Spain, ISBN Hardcopy: 978-0-88986-691-1 / CD: 978-0-88986-692-8, pp. 112-117

[20] Oravec, M.:A Method for Feature Extraction from Image Data by Neural Network Vector Quantization, Proc. of the 6th International Workshop on Systems, Signals and Image Processing, June 2-4, 1999, Bratislava, Slovakia, pp. 73-76

[21] Bovik, C. A.: Handbook of Image and Video Processing 2nd edition. Elsevier Academy Press, 2005, pp. 397-409