FACE RECOGNITION BASED ON THE OPTIMAL COMBINATION OF NEURAL NETWORKS, EIGENFACES AND LEAST SQUARES MATCHING METHODS

Prof. Dr. Werner M. Lechner [1]

Fachhochschule Hannover
University of Applied Sciences and Arts
Computer Science Department
D-30441 Hannover, Postbox 920251, Germany

Abstract: Automatic face recognition increases the security standards at public places and border checkpoints. The picture inside the identification documents could widely differ from the face, that is scanned under random lighting conditions and for unknown poses. The paper describes an optimal combination of three key algorithms of object recognition, that are able to perform in real time. The camera scan is processed by a recurrent neural network, by a Eigenfaces (PCA) method and by a least squares matching algorithm. Several examples demonstrate the achieved robustness and high recognition rate.

Keywords: Biometry, Artificial intelligence, Eigenfaces, Face recognition, Least Squares Matching, Neural networks, Principal Component Analysis

1. INTRODUCTION

The performance of an automatic face recognition system could be measured by two complementory parameters, the False Acception Rate (FAR) and the False Rejection Rate (FRR). Depending on the specific application one of those rates is of major interest. If the value of the FAR rate reaches 10%, then 10% of wanted persons could pass the border. In reverse, a FRR of 10% would result in a value of 10% of false alarms, that stresses the checkpoint staff. Therefore both recognition rates should be as best as possible.

An interesting concept to gain the necessary recognition rates is based on the optimal combination of different recognition algorithms, that are characterized by a complementary error behavior. The three selected effective algorithms are:

- Recurrent Neural Networks
- Eigenfaces (PCA, [2])
- Least Squares Matching methods.

Those methods do not need the localization of eyes or any other pattern and therefore the bunch graph matching methods [2] are not considered here.

All the algorithms should run under nearly real time conditions, because the recognition process must be performed within one second. A PC equipped with a dual-processor board was chosen as development system for shared memory parallel programming with the OpenMP standard.

[1] email: werner.lechner@inform.fh-hannover.de
http://www.wernerlechner.de/research

2. NEURAL NETWORKS

Artificial Neural Networks could be successfully applied to the recognition of graphic objects. The mathematic structure of an artificial neuron consists of two parts: The weighted sum of the input signals followed by a switching function, that maps that sum to the single output signal of the neuron. All the information is hidden in the numeric value of each single weight.

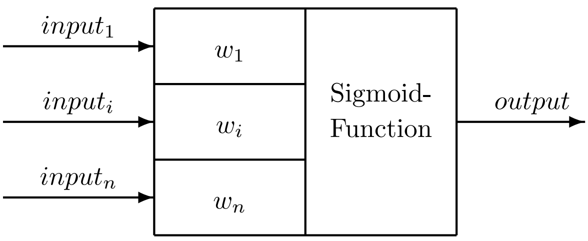

Fig. 1. Structure of an artificial neuron

Eq. (1) describes a single artificial neuron:

$$output = \text{sign}\left(\sum_{i=1}^{i=n}(w_i \cdot input_i)\right) \quad (1)$$

The Neural Network that is suitable for the sophisticated task of face recognition needs forward and backward connections as well as short cuts. Several topologies have been investigated. Best performance was gained with a topology presented by J. Hopfield [1].

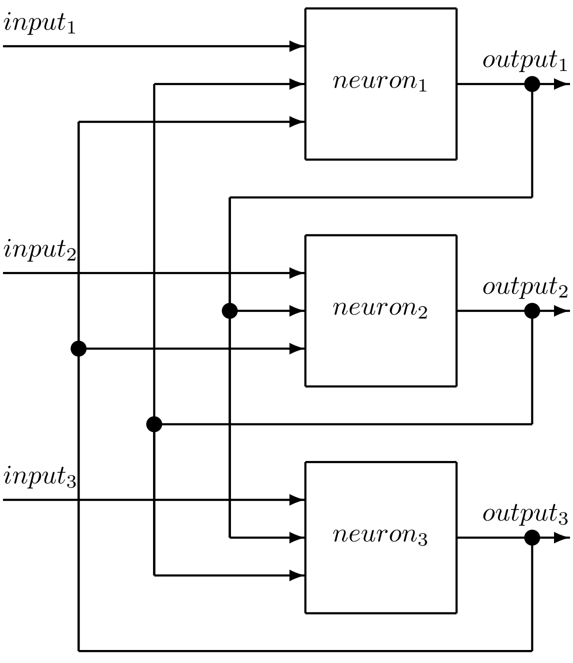

Fig. 2. Topology of a Hopfield Neural Network

The Hopfield Network is a fully connected network, but the neural output to itself is forbidden. This condition offers stability and a fast convergence for the recall phase. However, the big advantage of a Hopfield Neural Network consists in the possibility of an analytical calculation of the symmetric weights. With N neurons and M reference patterns P, the weights are defined by the Eq. (3) and (2):

$$w_{i,j} = \frac{1}{N}\sum_{m=1}^{m=M}(P_{m,i} \cdot P_{m,j}) \quad (2)$$

$$i = 1, N;\; j = i+1, N$$

$$w_{i,i} = 1 \quad w_{i,j} = w_{j,i} \quad (3)$$

There is a total number of N^2 weights w. Eq. (3) is a consequence of Eq. (2), because the sequence of the multiplication could be exchanged. The evaluation of Eq. (2) is performed only once at the initialization of the program. During the recall phase of the Hopfield Neural Network calculates a dynamic sequence of output signals that convergence after a few iteration steps.

Fig. 3 shows three examples of typical recall sequences of 60x80=4800 pixel pictures. The corresponding Hopfield Network consists of 4800 neurons and 4800x4800 = 23 millions of coefficients. Due to the symmetric weights only 11.5 millions of coefficients have to be calculated and stored in a down scaled Java Byte format.

The upper line of Fig. 3 shows a saluting person and how the hand is totally erased after three steps of iteration. The picture on the right side of the vertical black line is the reference picture stored in the weights of the Hopfield Network.

The first picture in the middle line displays a women with glasses. The first step compensates for that unknown graphic objects. The output pixels of the Hopfield network could now be compared to the stored picture shown on the right side.

The bottom line of Fig. 3 is the most interesting result, because in the face on the left side is hardly to recognize even for a human. However, the powerful Hopfield Neural Network is able to withdraw the cowl and after three steps of iteration the calculated pixel matrix is rather similar to the stored reference picture shown right of the black line.

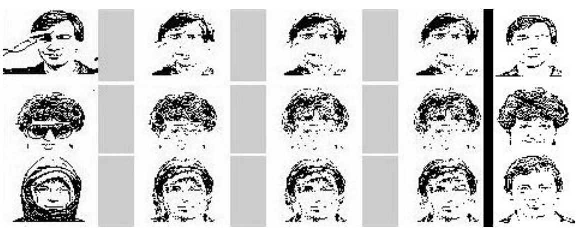

Fig. 3. Hopfield results for face recognition [5]

3. EIGENFACE METHOD

The method of the Eigenfaces was published as Principal Component Analysis (PCA) [3] and it applies the Eigenvalues of the linear algebra to the recognition of graphic objects. The serialized pixel matrix of a graphic object i corresponds to the i^{th} column of the matrix G. After deleting the mean value over the M pixel matrices, the matrix D is multiplied with its transpose in order to get the quadratic correlation matrix C.

$$g_i = (g_{11}, \ldots, g_{1m}, \ldots, g_{n1}, \ldots, g_{nm})^T \quad (4)$$

$$g_{mean} = \frac{1}{M}\sum_{i=1}^{M} g_i \quad \rightarrow \quad d_i = g_i - g_{mean} \quad (5)$$

$$C = DD^T = \begin{pmatrix} \vdots & \vdots & \vdots \\ d_1 & \ldots & d_M \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} \ldots & d_1 & \ldots \\ \vdots & \vdots & \vdots \\ \ldots & d_M & \ldots \end{pmatrix} = \begin{pmatrix} \sigma_{11}^2 & \ldots & \sigma_{1N}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{N1}^2 & \ldots & \sigma_{NN}^2 \end{pmatrix} \quad (6)$$

The matrix C reaches the order of N, which is too high for the calculation of the N Eigenvalues. A smart mathematic trick suggests to reverse the sequence of the matrix multiplication in Eg. (6), because the order the new matrix L is limited to the relativ small amount M of the available reference pictures.

$$L = D^T D = \begin{pmatrix} \ldots & d_1 & \ldots \\ \vdots & \vdots & \vdots \\ \ldots & d_M & \ldots \end{pmatrix} \begin{pmatrix} \vdots & \vdots & \vdots \\ d_1 & \ldots & d_M \\ \vdots & \vdots & \vdots \end{pmatrix} = \begin{pmatrix} \sigma_{11}^2 & \ldots & \sigma_{1M}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{M1}^2 & \ldots & \sigma_{MM}^2 \end{pmatrix} \quad (7)$$

The Eigenvalues of the matrix C could be gained from the M Eigenvalues of the matrix L:

$$Le_i = \lambda_i e_i \;\rightarrow\; D^T De_i = \lambda_i e_i \quad (8)$$

$$DD^T De_i = \lambda_i De_i \;\rightarrow\; CDe_i = \lambda_i De_i \quad (9)$$

The first M Eigenvectors v_i are collected within a new matrix V, that is named the Eigenface matrix and it consists of N lines and M columns. Each column displays an Eigenface. A scanned picture is transformed in the Facespace and compared with all the precalculated and stored reference Eigenfaces.

$$v_i = De_i \quad \longrightarrow \quad V = D\,[e_1, \cdots, e_M] \quad (10)$$

$$w_i = V^T(g_i - g_{mean}) = V^T d_i;\; i = 1, M \quad (11)$$

$$w_x = V^T(g_x - g_{mean}) = V^T d_x \quad (12)$$

$$V = \begin{pmatrix} v_{11} & \cdots & v_{1M} \\ \vdots & \ldots & \vdots \\ v_{N1} & \cdots & v_{NM} \end{pmatrix} \quad \text{with } N >> M \quad (13)$$

In order to receive a relative recognition error the maximal difference r_{ij} between all the vectors w_i could be calculated:

$$\Delta w_{i,j} = w_i - w_j \quad (14)$$

$$r_{i,j} = \sqrt{\Delta w_{i,j}^T \cdot \Delta w_{i,j}}; \quad i, j = 1, M \quad (15)$$

$$r_{max} = \text{maximum}(r_{i,j}) \quad (16)$$

$$r_{i,x} = \sqrt{\Delta w_{x,i}^T \cdot \Delta w_{x,i}}; \quad i = 1, M \quad (17)$$

$$\mu_{i,x} = \left(1 - \frac{r_{i,x}}{r_{max}}\right) 100\%; \quad i = 1, M \quad (18)$$

The relative value $\mu_{i,x}$ could be interpreted:

- if $\mu_{i,x} = 100\%$, then the picture i is identical to picture x.
- if $0 \leq \mu_{i,x} \leq 100\%$, then the picture i is member of the Facespace.
- If $\mu_{i,x} \leq 0$ for all i, then the picture x does not not belong to the Facespace.

As an example the Fig. 4 demonstrates the recognition of game cards. The upper line displays the four reference pictures. The first line ends with a scanned and unknown modification of the first game card. That card is rotated, scaled and scanned with a coloured background. The four Eigenfaces are shown in the second line. Those Eigenfaces correspond to the first four Eigenvalues of the matrix C and they are identical to the first four columns of the matrix V. At the bottom line the relative correlation values are visualized as thick vertical lines. The game card could be recognized and displayed on the right end of the second line with a correlation of 87%. The other games cards show much lower values of 57%, 21% and 13%. If the scanned picture does not belong to that game, the correlation would have reached even negative values.

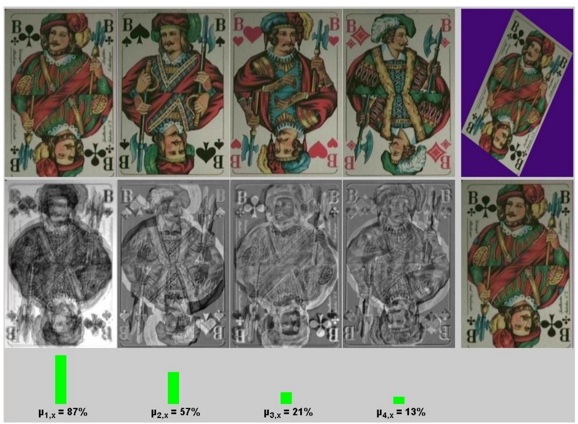

Fig. 4. Recognition of a game card

4. LEAST SQUARES MATCHING

Least squares matching [4] is applied to time series and it links a sequence of measured points by a smooth line on the basis of a least squares error. The algorithms could be extended to the evaluation of pixel matrices. For linear shifts Δx and Δy, for scaling factors of s_x and s_y and for a two-dimensional rotation with the angle α a geometric transformation matrix P could be defined on the basis of homogenios coordinates:

$$P = \begin{pmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} s_x\cos\alpha & -s_x\sin\alpha & \Delta x \\ s_y\sin\alpha & s_y\cos\alpha & \Delta y \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ 0 & 0 & 1 \end{pmatrix} \quad (19)$$

If the elements of the transformation matrix P would be known, then the points of each single reference picture R could be transformed to the picture Z and compared to the points of the scanned picture.

$$\begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} \\ p_{2,1} & p_{2,2} & p_{2,3} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{1,1} & \cdots & r_{1,m} \\ r_{2,1} & \cdots & r_{2,m} \\ 1 & \cdots & 1 \end{pmatrix}$$
$$= \begin{pmatrix} z_{1,1} & \cdots & z_{1,m} \\ z_{2,1} & \cdots & z_{2,m} \\ 1 & \cdots & 1 \end{pmatrix} \quad (20)$$

In order to reduce computing time under real time conditions a threshold value selects only a limited number of m significant points. The matrices R and P consists of three lines and of m columns.

Eq. (20) shows a matrix equation with unknown elements $p_{i,j}$. The Eg. (20) could be solved on the bases of a pseudo inversion by the least squares matching algorithm. The algorithm starts with a transpose operation, that exchanges the sequence of the matrix multiplication $P \cdot R$.

$$\begin{pmatrix} r_{1,1} & r_{2,1} & 1 \\ \vdots & \vdots & \vdots \\ r_{1,m} & r_{2,m} & 1 \end{pmatrix} \cdot \begin{pmatrix} p_{1,1} & p_{2,1} & 0 \\ p_{1,2} & p_{2,2} & 0 \\ p_{1,3} & p_{2,3} & 1 \end{pmatrix}$$
$$= \begin{pmatrix} z_{1,1} & z_{2,1} & 1 \\ \vdots & \vdots & \vdots \\ z_{1,m} & z_{2,m} & 1 \end{pmatrix} \quad (21)$$

A left-sided multiplication of the Eq. (21) with the Matrix R leads to the Eq. (22) :

$$\begin{pmatrix} r_{1,1} & \cdots & r_{1,m} \\ r_{2,1} & \cdots & r_{2,m} \\ 1 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{1,1} & r_{2,1} & 1 \\ \vdots & \vdots & \vdots \\ r_{1,m} & r_{2,m} & 1 \end{pmatrix} \cdot \begin{pmatrix} p_{1,1} & p_{2,1} & 0 \\ p_{1,2} & p_{2,2} & 0 \\ p_{1,3} & p_{2,3} & 1 \end{pmatrix}$$
$$= \begin{pmatrix} r_{1,1} & \cdots & r_{1,m} \\ r_{2,1} & \cdots & r_{2,m} \\ 1 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} z_{1,1} & z_{2,1} & 1 \\ \vdots & \vdots & \vdots \\ z_{1,m} & z_{2,m} & 1 \end{pmatrix} \quad (22)$$

The result of the multiplication between the matrices R and R^T is a quadratic matrix Q of the order of three.

$$Q = \begin{pmatrix} r_{1,1} & \cdots & r_{1,m} \\ r_{2,1} & \cdots & r_{2,m} \\ 1 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{1,1} & r_{2,1} & 1 \\ \vdots & \vdots & \vdots \\ r_{1,m} & r_{2,m} & 1 \end{pmatrix} \quad (23)$$

That quadratic matrix Q could easily be inverted. With the notation of Eq. (23) the Eq. (22) now looks like:

$$Q \cdot \begin{pmatrix} p_{1,1} & p_{2,1} & 0 \\ p_{1,2} & p_{2,2} & 0 \\ p_{1,3} & p_{2,3} & 1 \end{pmatrix}$$
$$= \begin{pmatrix} r_{1,1} & \cdots & r_{1,m} \\ r_{2,1} & \cdots & r_{2,m} \\ 1 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} z_{1,1} & z_{2,1} & 1 \\ \vdots & \vdots & \vdots \\ z_{1,m} & z_{2,m} & 1 \end{pmatrix} \quad (24)$$

Then the inverse of the matrix Q has to be calculated followed by a left-sided multiplication of Eq. (24).

$$Q^{-1} \cdot Q \cdot \begin{pmatrix} p_{1,1} & p_{2,1} & 0 \\ p_{1,2} & p_{2,2} & 0 \\ p_{1,3} & p_{2,3} & 1 \end{pmatrix}$$
$$= Q^{-1} \cdot \begin{pmatrix} r_{1,1} & \cdots & r_{1,m} \\ r_{2,1} & \cdots & r_{2,m} \\ 1 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} z_{1,1} & z_{2,1} & 1 \\ \vdots & \vdots & \vdots \\ z_{1,m} & z_{2,m} & 1 \end{pmatrix} \quad (25)$$

The product $Q^{-1}Q$ is equal to the identity matrix and therefore the unknown matrix P could be calculated by evaluation of the Eq. (26):

$$\begin{pmatrix} p_{1,1} & p_{2,1} & 0 \\ p_{1,2} & p_{2,2} & 0 \\ p_{1,3} & p_{2,3} & 1 \end{pmatrix}$$
$$= Q^{-1} \cdot \begin{pmatrix} r_{1,1} & \cdots & r_{1,m} \\ r_{2,1} & \cdots & r_{2,m} \\ 1 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} z_{1,1} & z_{2,1} & 1 \\ \vdots & \vdots & \vdots \\ z_{1,m} & z_{2,m} & 1 \end{pmatrix} \quad (26)$$

The error value is based on the difference between the picture matrices R_i and R_j and limited to the first and second component:

$$\delta x_i = R_i(k,1) - R_j(k,1) \quad (27)$$
$$\delta y_i = R_i(k,2) - R_j(k,2) \quad (28)$$
$$d_i = \sum_{k=1}^{k=m} \left[\delta x_i^2 + \delta y_i^2\right] \quad (29)$$
$$R_j = R_{imin} \quad (30)$$

The mininum of d_i corresponds to the index $imin$.

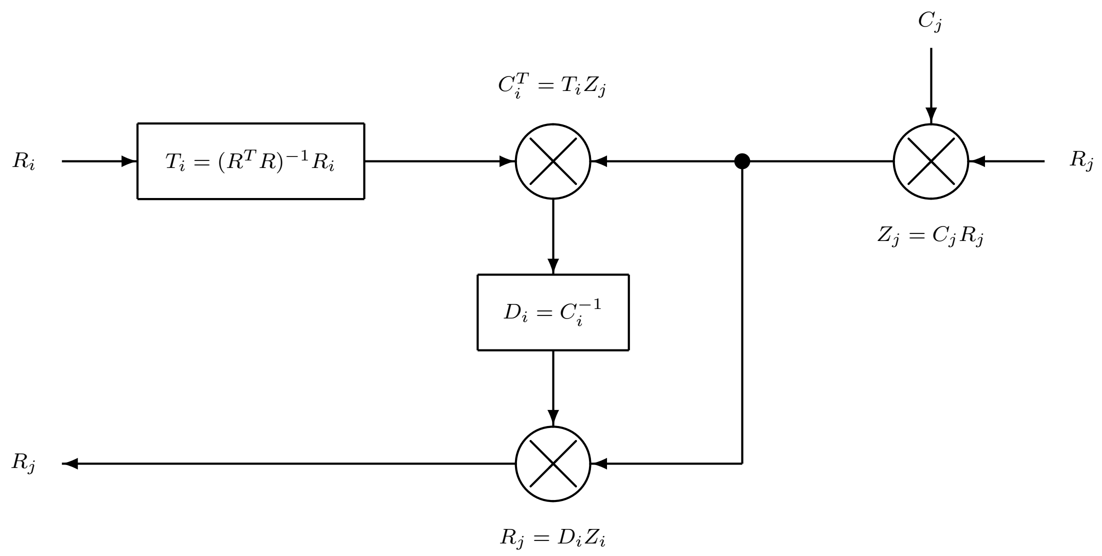

Fig. 5. Structure of the regression algorithms

The algorithm that is described in the Eq. (31), (32) und (35) consists out of several single steps and is visualized in Fig. 5. There are to solve the following system of equations:

$$T = (R^T R)^{-1} R \tag{31}$$

$$(T \cdot Z_{Scan}) \cdot R = Z_{Referenz} \tag{32}$$

$$\Delta x_i = Z_{Scan}(i,1) - Z_{Ref}(i,1) \tag{33}$$

$$\Delta y_i = Z_{Scan}(i,2) - Z_{Ref}(i,2) \tag{34}$$

$$s_i = \sqrt{\frac{1}{m} \sum_{i=1}^{i=m} (\Delta x_i^2 + \Delta y_i^2)} \tag{35}$$

In the initializing part of the algorithm the matrices T_i for each single reference picture R_i has to be calculated and applied in order to receive the matrices Z_i. The scanning camera performs a unknown transformation C_j. The comparison of the pictures is performed within the coordinate system of the reference pictures R.

The least squares matching performs best with noisy signals. Fig. 6 demonstrates on the basis of calculated faces the principle of the least squares matching method. The upper line shows randomly simulated pictures. One of those pictures is now selected, rotated, shifted, scaled and with a superposed noise level of 50% plotted below the original picture.

The least squares matching algorithms transforms back the modified picture and calculates the error between each single picture and the transformed one. The values of those errors are visualized as thick vertical lines in the bottom line of Fig.6. For the selected picture there is no error value. Each single transformed picture is plotted together with the given picture. Although the randomly scaled transform was unknown to the least square matching algorithm, the selected picture matches exactly the original picture. The next error rate is followed by a value of 40%. The human recognition system fails to recognize the right picture.

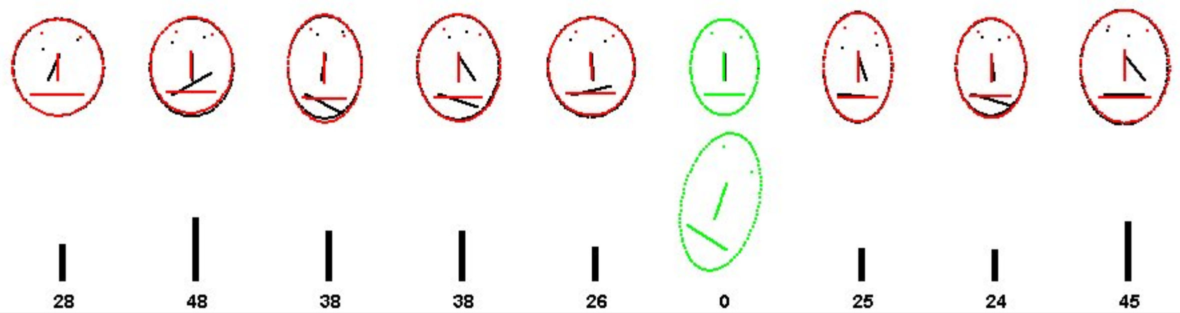

Fig. 6. Matching random pictures

5. OPTIMAL ALGORITHM COMBINATION

The presented algorithms are characterized by advantages as well as disadvantages (Tab. 1). If, for example, the scanned picture shows a face with glasses, the Neural Network is able to compensate for those unknown graphic objects. However, if the scanned object does not be a face at all, then the Neural Network fails, because it would deliver a face, that fits best to one of the stored faces.

The Eigenface method calculates in that case of a non face picture scan a negative error value which allows to distinguish between faces and non faces.

The least squares matching methods compensates superposed noise components, but runs into problems, if the scanned face is not similar to the stored face.

	advantage	disadvantage
Neural network	Compensates for unknown graphic objects	Traps on non face graphic objects
Eigenface method	Capability to distinguish between faces and non faces	Low capability to distinguish between faces
Least Square Matching	Compensates for noise that is added to the scan	Sensitive to variations of a face

Tab.: 1. Advantages and disadvantages

The optimal combination of those three methods is performed as shown in Fig. 7 and implemented on PC with a dual-processor mainboard with the JAVA OpenMP shared memory programming paradigma (JOPI) on the basis of the Windows XP operating system.

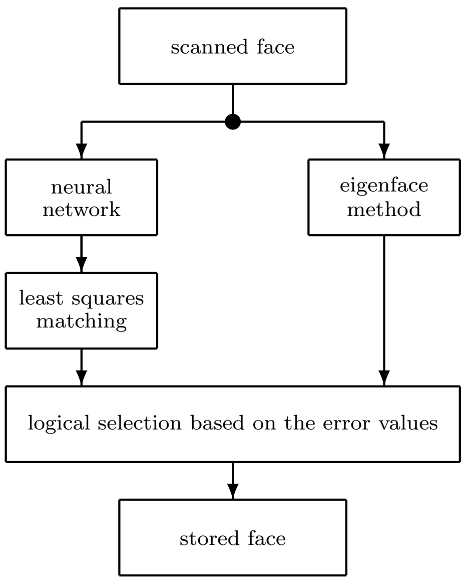

Fig. 7. Optimal combination of algorithms

The selection of the stored face that corresponds best to the scanned face is based on logical decisions taking into account the error values of each single algorithm. For a successful face recognition the μ-value of the Eigenface method must reach a positive value above 50% and a combined error value of all algorithms should not exceed 30%.

6. RESULT

Fig. 8 shows in the first upper line the 60x80 compressed pixel matrices of four different scans of the same person. In each case the recognition system selects the correct face and plots the corresponding reference picture below the scanned picture. In the lower picture blocks of Fig. 8 further examples of picture scans are shown. The Fig. 8 demonstrates a successful recognition for faces in different poses and sizes as well as for faces with added unknown graphic objects and backgrounds.

Fig. 8. Recognition results

7. REFERENCES

(1) Hopfield J., Tank D. (1986): *Computing with Neural Circuits: AModel*, Science, vol. 233, pp. 625-633

(2) Wiskott L., Fellous J., Krger N., Malsburg Ch. (1999): *Face Recognition by Elastic Bunch Graph Matching*, Intelligent Biometric Techniques in Fingerprint and Face Recogition, CRC Press, ISBN 0-8493-2055-0, Chapter 11, pp. 355-396

(3) Turk M., Pentland A. (1991), *Eigenfaces for recognition*, Journal of Cognitive Neuroscience, 3(1):71-86, 1991

(4) Bennamoun M., Mamic G. (2002): *Object Recognition, Fundamentals and Case Studies* Springer Verlag, ISBN 1-85233-398-7

(5) Manchester database, http:// peipa.essex.ac.uk/ipa/pix/faces/manchester