MATRIX METHODS IN PATTERN RECOGNITION

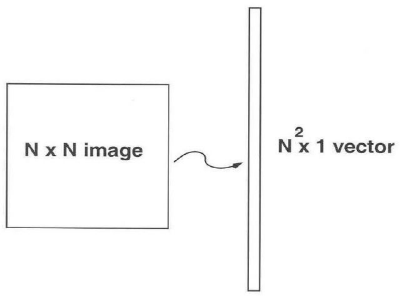
Elena Pelican and Lăcrămioara Grecu [1]
E-mails: {epelican, lgrecu}@univ-ovidius.ro

Abstract

Pattern recognition is a technique that tries to recognize data based on a priori knowledge or on a statistical information extracted from the patterns. It is "the act of taking in raw data and making action based on the category of the pattern"[2]. In this paper we shall exemplify the face recognition by presenting two approaches: the eigenfaces technique and the HOSVD one (Higher Order Singular Value Decomposition) for tensors. For each technique, we shall present some examples with the face and digit recognitions.

1 Introduction

In this section we shall briefly present the problem of *pattern recognition*. Given a image z (the picture of a face or a digit), we want to find out the closest image from the database $\{v_1, \ldots, v_M\}$. All the images have the same resolution (we shall consider the square resolution to simplify the exposure). Every single image is transformed into a vector. We created a face database of pictures of



11 persons, 10 pictures for each person and a digit database, with 10 pictures for each digit $\{0, 1, 2, \ldots, 9\}$.
The simplest thing we can do is to compare z with each v_i from the database. Thus, we want to find out an index $i_0 \in \{1, \ldots, M\}$ such that

$$\|z - v_{i_0}\| = \min_{1 \le i \le M} \|z - v_i\| . \tag{1}$$

In our experiments, for the consistent case of (1) (i.e. z belongs to the face or digit database), we obtained the results in Figure 1 below.

[1] "Ovidius" University of Constanta, Faculty of Mathematics and Computer Science
[2] R. O. Duda, P. E. Hart, and D. G. Storck, *Pattern Classification*, 2nd ed. Wiley-Interscience, New York, 2001

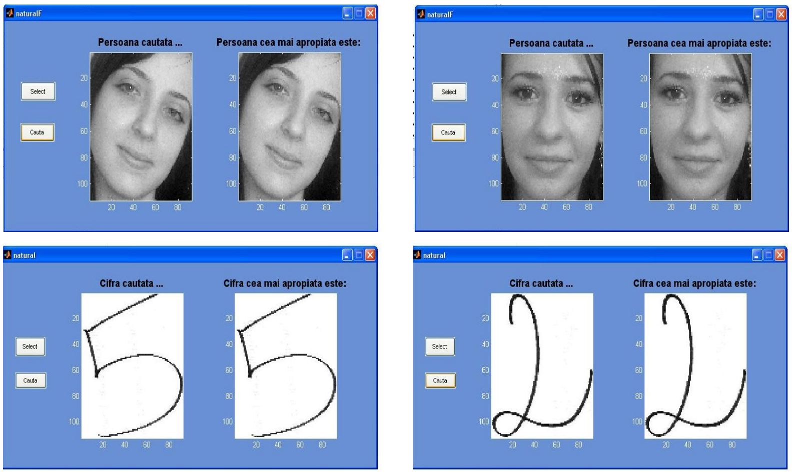


Figure 1: Consistent case of (1)

In the inconsistent case of (1) (i.e. z does not anymore belong to the coresponding database), we obtain unsatisfactory results, as one can see below in Figure 2.

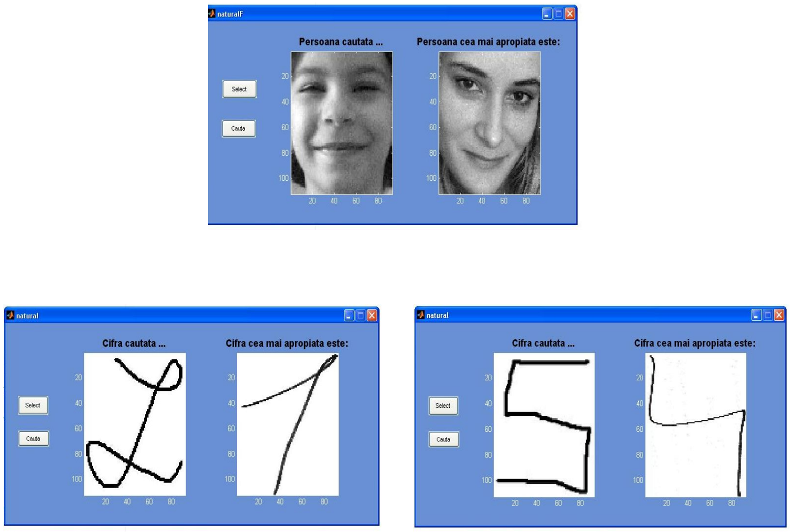


Figure 2: Inconsistent case of (1)

2 Eigenfaces Approach

2.1 Eigenfaces Algorithm

In *Eigenfaces algorithm* (proposed in [4]) we have to pursue the following steps.

- Set the database compossed by the images $I_1, I_2, \ldots, I_M$ (all images having the same resolution $N \times N$).
- Set every single image I_i as a vector Γ_i of $N^2 \times 1$ dimension.
- Compute the average face vector Ψ

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i.$$

- Substract the average face vector from all vectors

$$\Phi_i = \Gamma_i - \Psi.$$

- Compute the covariance matrix C

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T.$$

  where C of $N^2 \times N^2$ type and $A = [\Phi_1 \ \Phi_2 \ \ldots \ \Phi_M]$ is of $N^2 \times M$ type.
- Compute the eigenvectors q_i of $C = AA^T$.

For the matrix $C = AA^T$ it exists the orthogonal matrix Q such that $Q^T C Q = \text{diag}(\sigma_1, \ldots, \sigma_{N^2}) = D$, where $\sigma_i \geq 0$, $Q = \text{col}[q_1, \ldots, q_{N^2}]$ (see [1]). Since $Cq_i = \sigma_i q_i$ and $\{q_1, \ldots, q_{N^2}\}$ is an orthonormal basis in $\mathbb{R}^{N^2}$, any $x \in \mathbb{R}^{N^2}$ can be written as $x = \sum_{i=1}^{N^2} < x, q_i > q_i$.

Remark 1 *Similar considerations can be made for $\hat{C} = A^T A$.*

In practical applications, N^2 (the dimension of the matrix C) can be very large, thus the diagonalization $Q^T C Q = \text{diag}(\sigma_1, \ldots, \sigma_{N^2})$ can be very expensive. A possible solution is to keep from the N^2 eigenvectors $q_1, \ldots, q_{N^2}$, only the first K corresponding to the K largest eigenvalues $\sigma_1 \geq \ldots \geq \sigma_K > 0$. Because

$$Cx = \sum_{j=1}^{N^2} x_j C q_j = \sum_{j=1}^{N^2} \sigma_j x_j q_j \text{ and } Cx = \sum_{j=1}^{N^2} < x, \Phi_j > \Phi_j,$$

we have the approximation

$$Cx = \sum_{j=1}^{N^2} < x, \Phi_j > \Phi_j \approx \sum_{j=1}^{K} \sigma_j x_j q_j.$$

Now, we represent Φ_i in the basis $\{q_1, \ldots, q_{N^2}\}$.

$$\Phi_i = \Gamma_i - \Psi = \sum_{j=1}^{N^2} < \Phi_i, q_j > q_j.$$

Then we introduce the "truncated" vectors $\hat{\Phi}_i \approx \Phi_i$ as

$$\hat{\Phi}_i = \sum_{j=1}^{K} < \Phi_i, q_j > q_j = \sum_{j=1}^{K} w_j^i q_j \tag{2}$$

The vector $\hat{\Phi}_i$ is represented by the Fourier coefficients vector from (2), i.e.

$$w^i = \begin{bmatrix} w_1^i \\ w_2^i \\ \vdots \\ w_K^i \end{bmatrix} \in I\!R^K, \ i = 1, 2, \ldots, M.$$

Given a image Γ, with the same resolution as Γ_i, then we follow the steps

- Normalize Γ: $\Phi = \Gamma - \Psi$.
- Project on the eigenvectors space

$$\hat{\Phi} = \sum_{j=1}^{K} < \Phi, q_j > q_j = \sum_{j=1}^{K} w_j q_j.$$

- Represent $\hat{\Phi}$ as $w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}$.
- Find $i_0 \in \{1, \ldots, M\}$ satisfying $\|w - w^{i_0}\| = \min_{1 \le i \le M} \|w - w^i\|$.
- The corresponding database approximation for Γ is given by

$$\hat{\Gamma} = \sum_{j=1}^{K} w_j^{i_0} q_j. \tag{3}$$

Remark 2 *The algorithm works also for images with $N \times N_1$ resolution. In this case we obtain a vector Γ of dimension $NN_1 \times 1$.*

2.2 Experiments

In our experiments we have a face database with 110 pictures (92×112) and a digit database with 100 pictures (92×112). We considered K=20 because we have computed with Matlab the eigenvalues σ_j and noticed that σ_i, $i \geq 21$ were much smaller than the first 20 singular values of our matrix.
The results for the consistent case are the same as in Figure 1. For the inconsistent case we obtained the results from Figure 3, in which, for the digits pictures, they are much better than the ones from Figure 2.

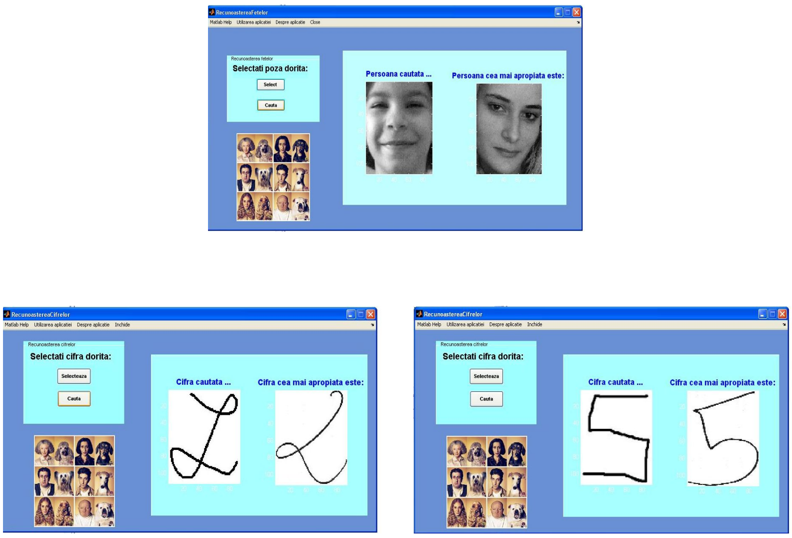


Figure 3: Inconsistent case

This means that the eigenfaces algorithm returns satisfactory results for the inconsistent case, unlike the similar case from Section 1.

3 HOSVD Approach

3.1 Preliminary results

In what follows, we shall briefly present a generalization of the matrix SVD theorem to 3-mode tensors (from now on, we shall call them just tensors), and afterwards some theorems which allows us to use the tensor SVD theorem for the problem of face recognition. For further details, see [3].

Let $A \in I\!R^{l \times m \times n}$, $U \in I\!R^{l_0 \times l}$, and $A \times_1 U$ a tensor of dimension $l_0 \times m \times n$, defined by

$$(A \times_1 U)(j, i_2, i_3) = \sum_{k=1}^{l} u_{j,k} a_{k,i_2,i_3}.$$

Similarly, we have 2-mode and 3-mode multiplication of a tensor by a matrix

$$(A \times_2 U)(i_1, j, i_3) = \sum_{k=1}^{m} u_{j,k} a_{i_1,k,i_3},$$

$$(A \times_3 U)(i_1, i_2, j) = \sum_{k=1}^{n} u_{j,k} a_{i_1,i_2,k}.$$

We can *unfold* a tensor into a matrix, $A_{(i)} = unfold_i(A)$, and we obtain

$$\begin{aligned} unfold_1(A) = A_{(1)} &= \left(A(:,1,:) \;\; A(:,2,:) \;\; \ldots \;\; A(:,m,:) \right), \\ unfold_2(A) = A_{(2)} &= \left(A(:,:,1)^T \;\; A(:,:,2)^T \;\; \ldots \;\; A(:,:,n)^T \right), \\ unfold_3(A) = A_{(3)} &= \left(A(1,:,:)^T \;\; A(2,:,:)^T \;\; \ldots \;\; A(l,:,:)^T \right). \end{aligned}$$

Using these unfoldings we obtain the following expresions for all three modes of multiplication:

$$\begin{aligned} A \times_1 U &= fold_1(U \cdot unfold_1(A)), \\ A \times_2 U &= fold_2(U \cdot unfold_2(A)), \\ A \times_3 U &= fold_3(U \cdot unfold_3(A)). \end{aligned}$$

One of the generalizations of the SVD theorem for tensors is the next one, often referred to as the higher order SVD, from now on denoted by HOSVD theorem (see [3]).

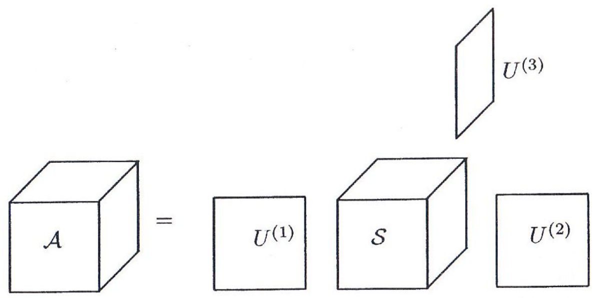


Figure 4: The SVD theorem for tensors

Theorem 1 *The tensor $A \in I\!R^{l\times m\times n}$ can be written as*

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$$

where $U^{(1)} \in I\!R^{l\times l}$, $U^{(2)} \in I\!R^{m\times m}$, $U^{(3)} \in I\!R^{n\times n}$ are orthogonal matrices. S is a tensor of the same dimensions as A and has the property that any two slices of S are orthogonal. The matrices $U^{(i)}$ result from $A_{(i)} = U^{(i)}\Sigma^{(i)}\left(V^{(i)}\right)^T$, $A_{(i)} = unfold_i(A)$.

We can write the HOSVD in different ways depending on what we want to do next. For example, we can write

$$A = D \times_e G \times_p H,$$

where $D = S \times_i F$. The e-mode multiplication is in fact the 2-mode multipli-

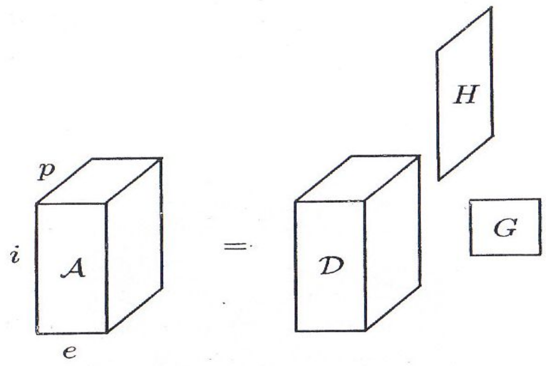


Figure 5: Another form of the SVD theorem for tensors

cation

$$(D \times_e G)\,(i_1, j, i_3) = \sum_{k=1}^{n_e} g_{jk} d_{i_1,k,i_3}.$$

If we set a parameter from the expresion with a particular value, for example, if we put $j = e_0$, it means that we only use the e_0^{th} row from matrix G. If we also put $i_3 = p_0$ we obtain

$$A\,(:, e_0, p_0) = D \times_e g_{e_0} \times_p h_{p_0},$$

where g_{e_0} is the e_0^{th} row from G and h_{p_0} the p_0^{th} from H.

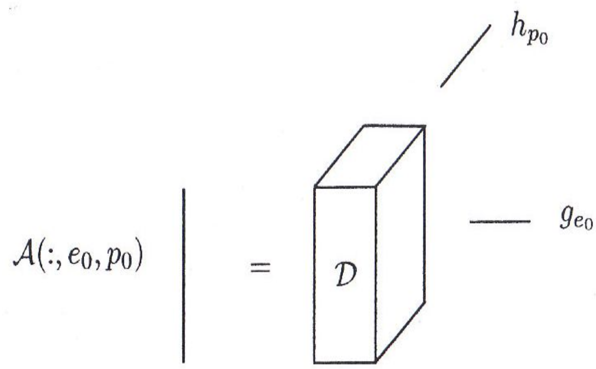


Figure 6: The obtainment of the person p_0 in the expression e_0

3.2 Face Recognition Using HOSVD

Given a picture of an unknown person, represented as a vector from $\mathbb{R}^{n_i}$, we want to decide if this person is or is not in our database, and if the answer is positive, to find out which of the n_p persons from our face database it represents. For this problem, we use the following form of the HOSVD of tensor A

$$A = C \times_p H, \ C = S \times_i F \times_e G$$

For a particular expression e we have

$$A(:, e, :) = C(:, e, :) \times_p H.$$

Tensors $A(:, e, :)$ and $C(:, e, :)$ are in fact matrices, which we can denote them by A_e and C_e, respectively. Hence we obtain the linear relations

$$A_e = C_e H^T, \ e = 1, 2, \ldots, n_e.$$

The same orthogonal matrix H appears in all n_e relations. If $H^T = \left(h_1 \ldots h_{n_p}\right)$, we get $a_p^{(e)} = C_e h_p$. Let $z \in \mathbb{R}^{n_i}$ be the picture of an unknown person, in an unknown expresion and we want to decide if the picture belongs to a person from the database or not. We can do that by computing its coordinates in all expresion bases and verify if the coordinates are the same (or almost the same) with the elements of a row from H. The coordinates of z in the e expresion basis can be obtained resolving the least squares problem

$$\min_{\alpha_e} \|C_e \alpha_e - z\|_2 . \tag{4}$$

The algorithm (see [2]) is
Algorithm A1
for $e = 1, 2, \ldots, n_e$

$$\begin{aligned}
&\quad \textit{solve } \min_{\alpha_e} \|C_e \alpha_e - z\|_2 \\
&\quad \textit{for } p = 1, 2, \ldots, n_p \\
&\qquad \textit{if } \|\alpha_e - h_p\|_2 < tol, \textit{ then is person } p \textit{ and } \text{STOP} \\
&\quad \textit{end}
\end{aligned}$$

end

For each image z we have to solve n_e least square problems with $C_e \in \mathbb{R}^{n_i \times n_p}$. This will take a lot of time to compute. From $C = S \times_i F \times_e G$ we obtain $C_e = F B_e$, where $B_e \in \mathbb{R}^{n_e n_p \times n_p}$, $B_e = (S \times_e G)(:, e, :)$. Matrix $F \in \mathbb{R}^{n_i \times n_e n_p}$ and we shall enlarge it so that it becomes square and orthogonal: $\hat{F} = \left(F \ F^{\perp}\right)$. We insert $\hat{F}^T$ inside the norm

$$\|C_e\alpha_e - z\|_2^2 = \left\|\hat{F}^T\left(FB_e\alpha_e - z\right)\right\|_2^2 = \left\|\begin{pmatrix} B_e\alpha_e - F^T z \\ -\left(F^{\perp}\right)^T z \end{pmatrix}\right\|_2^2 =$$
$$\left\|B_e\alpha_e - F^T z\right\|_2^2 + \left\|\left(F^{\perp}\right)^T z\right\|_2^2.$$

Hence we can solve the n_e least squares problems by solving

$$\min_{\alpha_e}\left\|B_e\alpha_e - F^T z\right\|_2, \; e = 1, 2, \ldots, n_e. \tag{5}$$

The algorithm is the following (see [2]).

Algorithm A2

Compute the QR decompositions of all matrices B_e, $B_e = Q_e R_e$, with $e = 1, 2, \ldots, n_e$.

  Compute $\hat{z} = F^T z$

  for $e = 1, 2, \ldots, n_e$

    Solve $R_e\alpha_e = Q_e^T\hat{z}$ *for* α_e

    for $p = 1, 2, \ldots, n_p$

      If $\|\alpha_e - h_p\|_2 < tol$, *then is person* p *and STOP*

    end

  end

Because the tensors and matrices have large dimensions, we can truncate them in such way that the truncated HOSVD can still approximate the tensor A. We can define $F_k = F\left(:, 1:k\right)$, for a value k that is much smaller than n_i, but larger than n_p. We obtain $\hat{C} = \left(S \times_e G\right)\left(1:k, :, :\right) \times_i F_k$. Hence we have to solve the least squares problems

$$\min_{\alpha_e}\left\|\hat{C}_e\alpha_e - z\right\|_2. \tag{6}$$

Also, we shall have $\hat{C}_e = F_k\hat{B}_e$, where $\hat{B}_e \in I\!R^{k\times n_p}$, and we obtain

$$\left\|\hat{C}_e\alpha_e - z\right\|_2^2 = \left\|\hat{B}_e\alpha_e - F_k^T z\right\|_2^2 + \left\|\tilde{F}_{\perp}^T z\right\|_2^2,$$

with $\hat{z}_k = F_k^T z$.

3.3 Experiments

With the same databases from Section 2, we implemented both **A1** and **A2** algorithms. The results for both of them were the same, but the algorithm **A2** is faster than the **A1**. For the consistent case, the results are those depicted in Figure 7, while for the inconsistent case, are those from Figure 8.

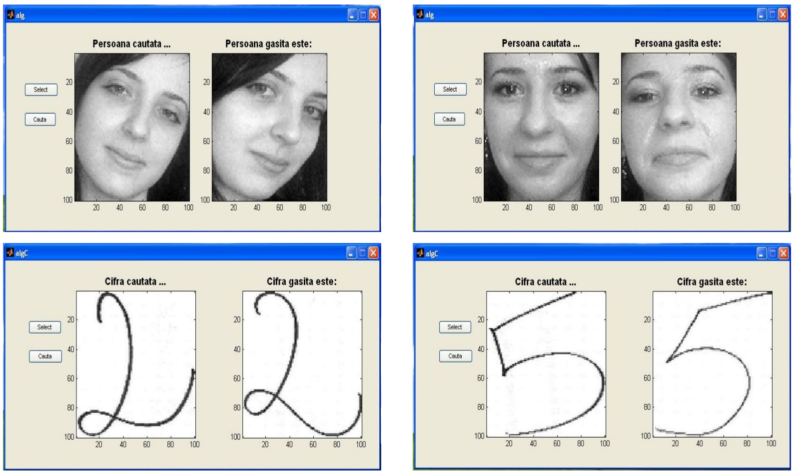


Figure 7: Consistent case

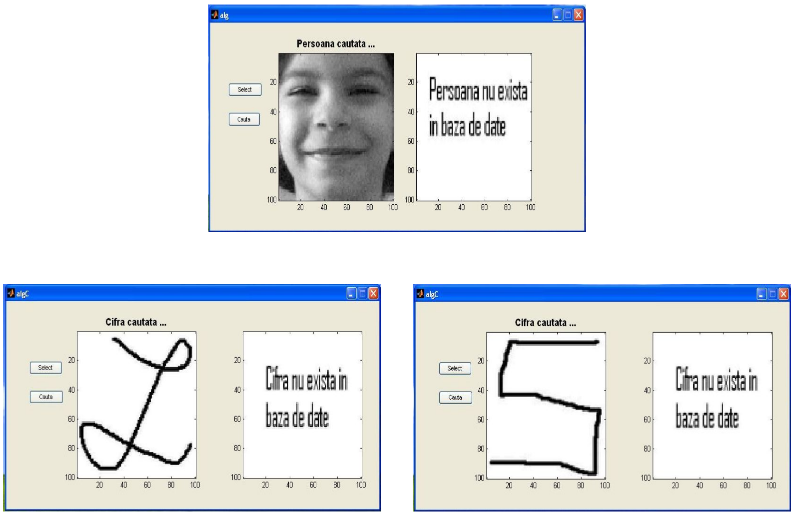


Figure 8: Inconsistent case

This means that this approach returns satisfactory results for the consistent case. For the inconsistent case, it eliminates the human decision factor in face recognition, but in digit recognition there is a drawback because it does not find the closest digit.

4 Conclusions and Future Work

In the consistent case we know from the very beginning that the pattern we are looking for (face or digit) is in our database, so both approaches have satisfactory results. In the real life applications, the problems are generally inconsistent. So, for face recognition is better to use the HOSVD approach because we do not need the human decision factor. But for digit recognition,

we shall use the Eigenfaces approach, because it returns the closest digit to the one we are looking for, and this could be helpfully if we want to decode a handwritten postal code or phone number, and so on.
As future work, we want to improve the running time for the tensor algorithm and to enlarge our digit database, so the program can return the results faster. We also want to use the Eigenfaces technique and wavelets analysis with respect to the face detection, and edge detection, respectively.

REFERENCES

1. A. Björck, *Numerical Methods For Least Squares Problems*, SIAM, 1996.
2. L. Elden, *Matrix Methods in Data Mining and Pattern Recognition*, SIAM, Philadelphia, 2007.
3. L. De Lathauwer, B. de Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21:1252–1278, 2000.
4. M. Turk and A. Pentland, *Eigenfaces for Recognition*, Journal of Cognitive Neuroscience, vol. 3, no. 1, 1991.
5. http://openbio.sourceforge.net/resources/eigenfaces/eigenfaces-html/facesOptions.html.
6. http://www.cs.ucsb.edu/ mturk/Papers/mturk-CVPR91.pdf.