



**HAL**  
open science

## Face localization by neural networks trained with Zernike moments and Eigenfaces feature vectors. A comparison

Mohammed Saaidia, Anis Chaari, Sylvie Lelandais, Vincent Vigneron, Mouldi Bedda

► **To cite this version:**

Mohammed Saaidia, Anis Chaari, Sylvie Lelandais, Vincent Vigneron, Mouldi Bedda. Face localization by neural networks trained with Zernike moments and Eigenfaces feature vectors. A comparison. 2007 IEEE International Conference on Advanced Video and Signal based Surveillance, Sep 2007, London, United Kingdom. 6 p., 10.1109/AVSS.2007.4425340 . hal-00147392

**HAL Id: hal-00147392**

**<https://hal.science/hal-00147392>**

Submitted on 16 May 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Face localization by neural networks trained with Zernike moments and Eigenfaces feature vectors. A comparison

M. Saaidia

A. Chaari, S. Lelandais, V. Vigneron

M. Bedda

Université Evry Val d'Essonne  
Evry-courcouronnes, FRANCE  
saaidia@iup.univ-evry.fr

Université Evry Val d'Essonne  
Evry-courcouronnes, FRANCE  
{chaari, vvigne}@iup.univ-evry.fr  
s.lelandais@iut.univ-evry.fr

Université de Annaba  
Annaba, Algérie  
mouldi\_bedda@yahoo.fr

## Abstract

Face localization using neural network is presented in this communication. Neural network was trained with two different kinds of feature parameters vectors; Zernike moments and Eigenfaces. In each case, coordinate vectors of pixels surrounding faces in the images were used as target vectors on the supervised training procedure. Thus, trained neural network provides on its output layer a coordinates vector  $(R, \theta)$  representing pixels surrounding the face contained in treated image. This way to proceed gives accurate faces contours which are well adapted to the faces shapes. Performances obtained for the two kinds of training feature parameters were recorded using a quantitative measurement criterion according to experiences carried out on the XM2VTS database.

## 1. Introduction

In the last two decades, face detection received a growing attention by the researchers concerned with Human-Machine communication. Thus, many face detection methods and algorithms were developed for images and video which try to overcome different constraints like difference in brightness, pose and movement, faces appearance (glasses, beard, moustache), execution time, etc... . These methods were useful for more complex techniques in Human-Machine communication like face recognition (identity check), gesture communication and face expression analysis and recognition.

According to Hjelmas and Low [1], face detection methods can be classified as "global approach" which consists in entirely seeking face and "components approach" which consists in finding the face through localization and regrouping of its components (eyes, nose...). They can be also classified according to face characteristics used like color, shape and movement.

The two methods presented and compared in this work can be classified as global methods. These two methods differ only on the first step which is the way to characterize the image to be treated. The first exploits geometrical characteristics of the face and the second uses projection on image sub-space variations. In the second step, a neural network, beforehand trained, uses the feature vector produced in the first step to deliver on its output layer a coordinates vector for pixels of the face's probable contour contained in the treated image. To make objective measure and comparison of methods performances we use a quantitative measurement criterion [2].

Geometrical moments, particularly Zernike ones, are used here for their capacity to compress the geometrical information, contained in the image treated, in a rather reduced parameters vector by projection of the image on an orthogonal basis [3]. In the same way, the Eigenfaces characterize the image by a reduced parameters vector representing variations of the treated image around an average image and according to some variation directions [4]. This compression characteristic makes them very adapted to the training of classifiers, like neural networks, who often need, on their input layer, feature vectors reduced in size but rather representative of the element subject to the classification. Zernike moments were particularly used for face recognition [5], [6] and target recognition in general [7]. Eigenfaces were largely used in face detection and recognition directly [4], [8] or with neural networks [9].

In the following of this communication we will explain Zernike moments and Eigenfaces formulations in section 2 then, in section 3 we develop the proposed way to their practical implementation. In section 4 we will expose the measurement criterion and in section 5 methods performances results. Finally, section 6 will contain the conclusion.

## 2. Zernike moments and Eigenfaces Formulation

### 2.1. Zernike moments

Zernike moments are part of the geometrical moments general theory. They were introduced initially by F. Zernike. Zernike moments are built on a set of orthogonal polynomials which allow construction of orthogonal base given by Eq. (1).

$$V_{n,m}(x,y) = V_{n,m}(\rho, \theta) = R_{n,m}(\rho) \cdot e^{j \cdot m \cdot \theta} \quad (1)$$

where:

$$\left\{ \begin{array}{l} R_{n,m}(\rho) = \sum_{k=|m|}^n \frac{(-1)^{(n-k)/2} \cdot (n+k)!}{\left(\frac{n-k}{2}\right)! \left(\frac{k+m}{2}\right)! \left(\frac{k-m}{2}\right)!} \cdot \rho^k \\ \rho = \sqrt{x^2 + y^2} \quad \text{and} \quad \theta = \arctg(y/x) \end{array} \right.$$

with:  $n \geq 0$ ,  $m \neq 0$ ,  $m < n$ ,  $n - m < n$  and  $(n-k)$  even.

$R_{n,m}(\rho)$  are the orthogonal radial polynomials,  $\mathbf{n}$  is the order of the moment and  $\mathbf{m}$  the repetition factor (the smoothness of the required details) at this order.  $\rho$  and  $\theta$  are respectively the radius and the angle of function's treated point.

This base being orthogonal only inside the unit circle, the image to be projected must be mapped according to Eq.(2) which gives relations between the relative coordinates  $(\mathbf{i}, \mathbf{j})$  of the initial image pixels and the new pixels coordinates  $(x_j, y_i)$  of the mapped one.

$$x_j = c + \frac{j \cdot (d - c)}{(Q - 1)} \quad \text{and} \quad y_i = d - \frac{i \cdot (d - c)}{(P - 1)} \quad (2)$$

With  $(\mathbf{P}, \mathbf{Q})$  dimensions of the image to be projected,  $\mathbf{i}$  and  $\mathbf{j}$  indices of the point to be mapped and  $(\mathbf{c}, \mathbf{d})$  couple of parameters allowing to map the function inside the unit circle (completely:  $(c=-1/\sqrt{2}$  and  $d=c$  or partially:  $c=-1$  and  $d=1$ ). The projection of a numerical function, in general,  $f(x_j, y_i)$  on the basis functions of Eq.(1) gives the Zernike moments  $Z_{n,m}$  according to Eq.(3).

$$Z_{n,m} = \frac{n+1}{\pi} \sum_{x_j^2 + y_i^2 \leq 1} f(x_j, y_i) \cdot V_{n,m}^*(x_j, y_i) \quad (3)$$

where: \* denotes the complex conjugate of the function.

Traditional formulation of Zernike moments is very easy to implement but its computational time cost is very high. Researchers tried to overcome this major handicap by developing new formulations to enhance the speed computation [10], [11]. The proposed algorithm in [11], which is adopted here, has the advantage to preserve the same accuracy of computation as in the traditional formulation. To lead to this form of representation, the previous equations are rewritten and reorganized as shown in Eq.(4) which reduces the Zernike moments computing of any image to the computing of a linear combination of  $\beta_{n,m,k}$  and

$X_{m,k}$ .

$$\begin{aligned} Z_{n,m} &= \frac{n+1}{\pi} \sum_{x^2 + y^2 \leq 1} \left( \sum_{k=|m|}^n \beta_{n,m,k} \cdot \rho^k \right) e^{-j \cdot m \cdot \theta} \cdot f(x_j, y_i) \\ &= \frac{n+1}{\pi} \sum_{k=|m|}^n \beta_{n,m,k} \cdot \left( \sum_{x^2 + y^2 \leq 1} e^{-j \cdot m \cdot \theta} \cdot \rho^k \cdot f(x_j, y_i) \right) \\ &= \frac{n+1}{\pi} \sum_{k=|m|}^n \beta_{n,m,k} \cdot X_{m,k} \end{aligned} \quad (4)$$

where: 
$$\beta_{n,m,k} = \frac{(-1)^{(n-k)/2} \cdot (n+k)!}{\left(\frac{n-k}{2}\right)! \left(\frac{k+m}{2}\right)! \left(\frac{k-m}{2}\right)!}$$

According to this formulation, we need only  $2 \cdot \left(\frac{n}{2} + 1\right)^2 + (P^2 - 1)$  additions and  $\frac{n^2 \cdot P^2}{2} + 2 \cdot P^2 \cdot n$  multiplications to compute Zernike moments up to order  $n$  for an image of  $(P \times P)$  pixels [11].

## 2.2. Eigenfaces

“Eigenfaces” was the first method successfully used for face treatments like face detection and face recognition [4]. This method is based on the decomposition of the treated image according to some directions of variation around an average image. Decomposition is performed on a set of representative images of the characteristics to be classified. Based on Principal Components Analysis (PCA), Eigenfaces method uses SVD operation (Singular Values Decomposition) on a matrix containing a set of vectors, representing images, to determine their principal variety directions. In the case of faces images, these main directions were called Eigenfaces.

To use Eigenfaces method, we first construct a projection space by operating SVD on the covariance matrix  $C_x$  given by Eq.(5). This operation gives the eigenvectors and eigenvalues of  $C_x$  arranged according to the variety directions importance.

$$C_x = (X - \mu_x) \cdot (X - \mu_x)^T \quad (5)$$

where:  $X = [I_1, I_2, \dots, I_L]$  is an  $(K \times L)$  face's matrix with  $L$  the number of faces and  $K = P \times Q$  the dimension of the face's vector  $I_i$ ,  $(\cdot)^T$  transpose matrix of  $(\cdot)$ , and  $\mu_x$  the average face :

$$\mu_x(j) = \frac{1}{L} \sum_{i=1}^K I_i(j).$$

The  $L$  faces images are selected to be well representative of characteristics variability. The set of eigenvectors obtained is used as projection space for images to be treated.

## 3. Methods implementation

Our contribution, with the goal to localize face in image, consists in three propositions:

The first one consists in the use of Zernike moments as training feature vectors for a neural network. Indeed, in addition of their capacity to include geometrical information of an image and to compress them in a reduced vector, Zernike moments are not abstract parameters. Each one of them have a significance related to the statistical characteristics of the image which they represent such as the surface, the total mass center, mass centers in horizontal and vertical directions, horizontal and vertical symmetry, ...etc. Thus, a face by its particular shape and its contents geometrically rich by the details of the elements which it contains (eyes, mouth, eyebrows...) will be well represented in the parameters of Zernike feature vector.

The second one is the manner in which methods implementation will be done. According to the previous works on face detection, we found that methods developed usually use

rectangular or elliptical windowing research of the face on the treated image. This procedure gives non precise faces contours and in some cases requires additional refinement operations. In our work we propose to train the neural network on target vectors which contain pixels coordinates obtained by manually delimiting faces in training images. This procedure will produce trained neural networks which provide precise and adapted faces contours according to their shapes.

The third proposition consists in the use of a quantitative measurement criterion to record and compare the results obtained by each one of the two implemented methods. The criterion is based on the compute of methods performances according to the number of pixels correctly and wrongly detected as belonging to the face in the treated image.

Figure 1 gives the block diagram of the proposed face localization system.

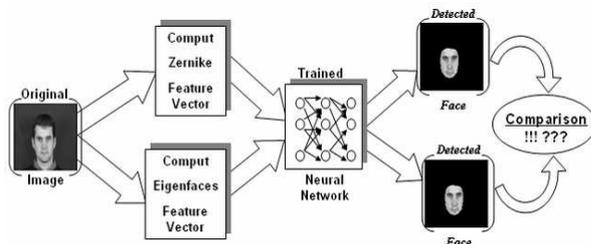


Figure 1 : General diagram of face localization system

It should be notified here, that we make no assumption on the probable shape of the face subject to detection and no pre-processing operation is required for the treated image. Implementation of our method is mainly based on training phase which will be achieved in four stages:

- Computation of Zernike moments and Eigenfaces vectors for all the  $N$  images of the work database.
- Construction of the training database by randomly taking  $N1$  images from the work database ( $N1 \ll N$ ) and their corresponding Zernike moments vectors  $Zi$  and Eigenfaces vectors  $Ei$ .
- Manual delimitation of the face area in each image of the training database by a set of points  $Ci$  representing its contour.
- Training of neural networks on the  $N1$  sets of couples  $(Zi, Ci)$  or  $(Ei, Ci)$ .

Neural networks trained with Zernike feature vectors learn to extract statistical information contained in Zernike moments and in there interactions which are closely related to the area of the required face. Those trained with Eigenfaces feature vectors learn to identify the main variety directions introduced by face in training images.

To test and measure the performances of the network obtained after training operation, we proceed, according to Figure 1, on all  $(N-N1)$  images remaining in the work database. Face localization procedure will be the same for the two methods compared in this work and it will be done in two steps:

- During the first step, an image is presented to an algorithm which extracts Zernike or Eigenfaces feature vector.

- At the second step, a back-propagation neural network, beforehand trained, receives on its input layer the feature vector which was computed in first step. In response, it gives on its output layer a coordinates vector for a set of points representing the probable contour of the face contained in the treated image.

Results obtained by each one of the methods according to equivalent parameters and for the same images are then quantitatively compared.

## 4. Quantitative measurement criterion

To give an objective appreciation of results given by the methods studied here, we propose a new way to calculate the detection rate based on the relation between the number of pixels correctly and wrongly detected as pixels of the face, the number of face pixels and the number of all pixels in the treated image. To do so, all the images of the testing database were manually segmented in three regions. The first region (white one on the masks of Figure 2) contains the  $W$  pixels which represents the essential components of the face (brows, eyes, nose, mouth and surrounding pixels). The second region (grey one on the masks of Figure 2) represents the pixels surrounding the first region and belonging to the face. The last region represents all the  $B$  pixels of the image which are not parts of the face. For the detection system, the first region is one which have to be contained imperatively in the resulting contour and the third one is to be imperatively discarded from it. The second region is optional and has no effect on the computed results.

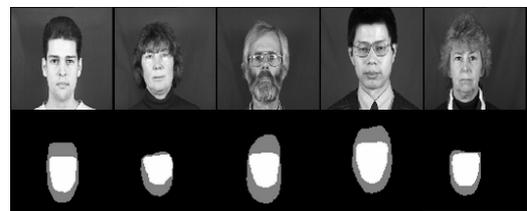


Figure 2: Examples of regions definition. Top: original image, Bottom: mask of regions

We define two types of rates; Good detection rate (**Gdr**) and Quality detection rate (**Qdr**)

$$Gdr = \frac{W1}{W} . 100 \quad \text{and} \quad Qdr = \left( \frac{W1}{W} - \frac{B1}{B} \right) . 100 \quad (6)$$

Where  $W1$  and  $B1$  are respectively the number of pixels correctly and wrongly detected as belonging to the face. The **Gdr** measures how many pixels from the essential parts of the face are detected. The **Qdr** gives a more strict measure of face detection taking pixels of images that wrongly detected as belonging to the face into account. These two rate measures are complementary. Having only **Gdr** we don't know how many pixels are wrongly detected as belonging to processed face. In the same way, having **Qdr** only we don't know how many pixels belonging to the face are correctly detected. On Figure 3 we illustrate this fact on some examples with recorded **Gdr** and **Qdr**.

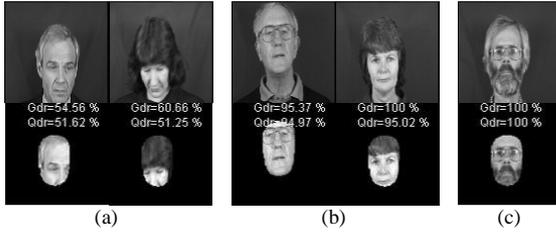


Figure 3: *Top: original images. Bottom: Difference between Gdr and Qdr for face localization*

On Figure 3.a we have the same bad Qdr (about 50%), with two different Gdr (55% and 60%). On Figure 3.b it's the same situation for a good Qdr (about 95%) with two different Gdr values (95% and 100%). To finish, we give on Figure 3.c an example of a face perfectly detected with Gdr and Qdr at 100%. Thus, to have a good appreciation of recorded results, each one of Gdr and Qdr have to be computed. Best results are obtained when they are both closest to 100% with the minimum difference between them.

## 5. Experimental results

In order to check the validity of our proposed method and to compare methods performances studied here, experimental studies were carried out on the XM2VTS images database [13]. It contains 4 recordings of 295 subjects taken over a period of 4 months with rotating head shot in vertical and horizontal directions. Images are color and in ppm format.

In our experiences we first brought some transformations to the original images like the change to gif format (more compressed) and the use of luminance information only (grey scale images) to compute the Zernike moments and Eigenfaces vectors.

To obtain the training database we take randomly 15 images of different people, each one with 3 different recordings, so that gives 45 example couples ( $Z_i, C_i$ ) and ( $E_i, C_i$ ) for training the neural network.

To have a precise and rather general idea on the performances of the method, we carried out the construction of 20 training databases always by randomly taking the examples. For each database, the network was trained then tested on the whole of the remaining images. For each test, we compute the average values of Gdr and Qdr and there Standard deviation (Std).

Our experiences aimed at the study of the behavior of the two methods with respect to the training database, the dimension of the training vectors and the complexity of the neural network. Neural networks trained and used in our experiences have 60 neurons on their output layers so they provide 30 coordinates pairs (R,θ) corresponding to 30 pixels surrounding the region supposed containing the face in the treated image.

### 5.1. General results

First, we present on Figure 4 an example of results obtained by applying the two methods for each one of the 295 images representing part of the testing database.

These rates were obtained on two different neural networks. The first was trained with Zernike moments feature vectors with

$n=5$  and  $m=1$  on the training database number 15 which allow to obtain the best results with feature dimension vectors equal to 22. The second was trained with Eigenfaces feature vectors with dimension 22 on the training database number 8 which also gives the best results for Eigenfaces at this dimension. For the two neural networks the hidden and output layers have respectively 10 and 60 neurons

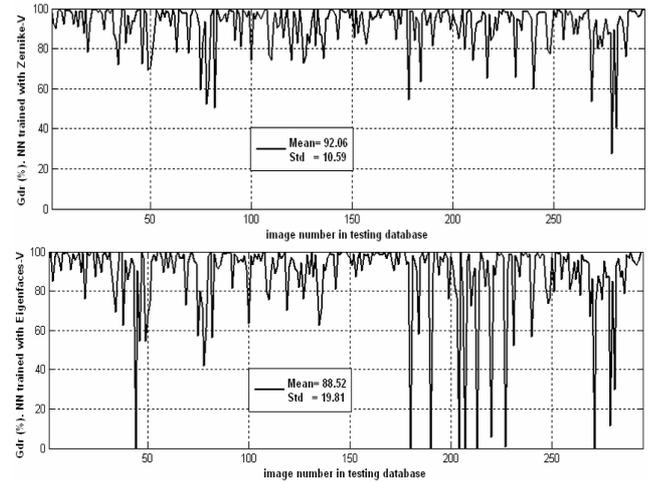


Figure 4: *Qdr recorded for 295 images of the testing database. Neural network trained with: (top) Zernike moments (Bottom) Eigenfaces*

The resulting rates show that in the case of Zernike moments only few faces were incorrectly detected. Most of the images were correctly treated indicating good generalization performances. However, less performance results were recorded in the case of Eigenfaces. In Table 1 we see that for the first case, Zernike moments training, 90% of images have Gdr and Qdr greater than 80%, however only 84% in the case of Eigenfaces. This performances superiority can be also seen by comparing the general Qdr's Mean and Std computed for all the images of the testing database.

Table 1: *Performances comparison for results reported in Figure 4 (Nbi: Number of images).*

	Zernike Training		Eigenfaces Training	
	Nbi	(Nbi/295) %	Nbi	(Nbi/295) %
Gdr<70%	11	3.73	23	7.80
Gdr<80%	23	7.80	37	12.54
Qdr<70%	13	4.41	27	9.15
Qdr<80%	32	10.85	47	15.93
Qdr Mean	92.06		88.52	
Qdr Std	10.59		19.81	

On Figure 5 we give some examples of good detected faces from the testing database. We chose images with some faces variability in terms of position, color, pose, size and gender. Results illustrate the difference between Gdr and Qdr measures and also between performances of the two compared methods.

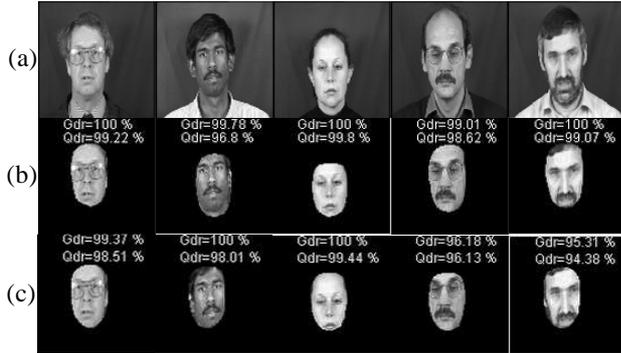


Figure 5 : Examples of face detection.(a): original image. (b): face detected with Zernike vectors training (c): face detected with Eigenfaces vectors training

## 5.2. Training database influence

To study training database influence on each one of the two methods presented and also to obtain more reliable performances comparison between them, experiences were carried out on the 20 training databases randomly constructed. Results given on Figure 6 were obtained by training, testing and measuring performances of a neural network for each one of the twenty training databases with the same fixed parameters. Input, hidden and output layers have respectively 6, 10 and 60 neurons and Sigmoid, Sigmoid and Linear as activation functions. "Resilient propagation" was used as neural training function.

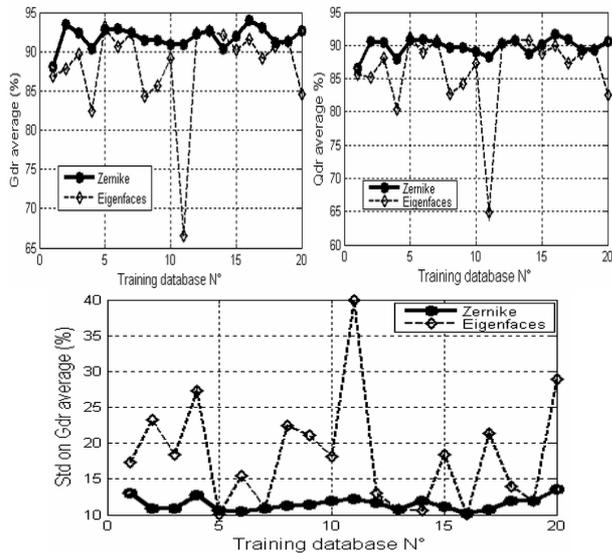


Figure 6: Gdr, Qdr and Std variations related to the Training feature vectors (Zernike or Eigenfaces) and databases (1,2, ..., 20). Top: Gdr and Qdr averages, Bottom: Std

According to Gdr, Qdr and Std reported by curves in Figure 6 we can say that training with Zernike feature vectors gives best results than training using Eigenfaces ones. Indeed, for the first

case Gdr averages are greater than 90% (up to 94%) in almost the totality of the training databases and the Qdr are about 90%. In the same way, low Std values show good generalization performances on the images of the testing database. The results obtained for Eigenfaces training show a greater sensitivity to the training databases and a bad generalization performance (big values of Std) for most of them. Indeed, like it is shown on Table 2, up to 23% of difference in Gdr and up to 40% in Std are recorded for neural networks trained with Eigenfaces feature vectors, according to the training database. This difference is only about 7% and Std values are no more than 14% for those using Zernike moments as training vectors.

Table 2 : Training databases influence

	Min Gdr	Max Gdr	Min Std	Max Std
Zernike Training	<b>86.84</b>	<b>94.01</b>	<b>10.13</b>	<b>13.56</b>
Eigenfaces Training	<b>66.55</b>	<b>93.12</b>	<b>10.09</b>	<b>39.95</b>

## 5.3. Feature vectors size influence

Feature vectors size has significance related to the quantity of image information included and compressed by these vectors. Zernike vectors size is controlled by parameters m and n while that of Eigenfaces is controlled by the size of the projection space constructed. Moreover, vectors size determines the number of neurons in the neural network input layer and hence, its complexity. On Figure 7 we give the variation curves of Gdr and Qdr averages and the Std, computed on the totality of the twenty training databases, according to 7 different size values of the feature vectors.

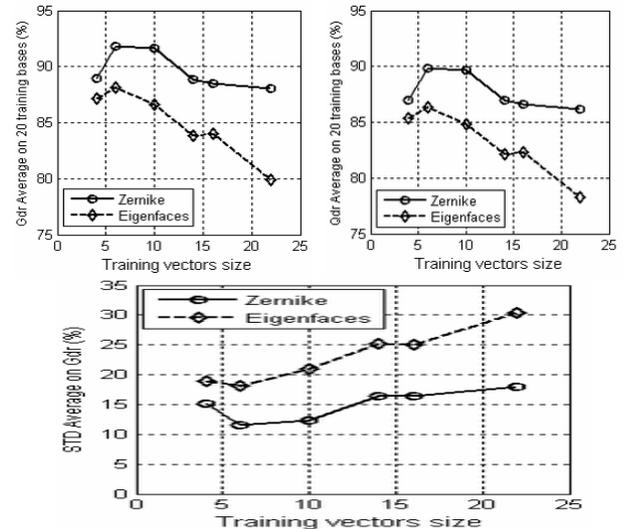


Figure 7: Gdr and Qdr averages and std variations computed on 20 independent training databases with 7 vectors size: 4 (n=1,m=1), 6 (n=2,m=1), 10 (n=3,m=1), 14 (n=4,m=2), 16 (n=5,m=3), 22 (n=5,m=1) and 24 (n=6,m=3)

Here also, we can see that Zernike moments vectors provide best average results for the 7 cases studied. For the two methods, best results were obtained for the vectors size 6. Decreasing evolution of Gdr and Qdr averages is observed for sizes greater than 10 where, in the same time, the Std values increase considerably. This indicates a decrease in the generalization capacity of the trained neural networks. Indeed, increasing vectors sizes increase the neural networks complexity which converge more difficultly.

## 6. Conclusion

Face localization using neural networks and a new way to train them were presented in this communication. We compared, results given by neural networks trained with Zernike moments feature vectors and those trained with Eigenfaces ones, according to a proposed quantitative measurement criterion which allows an automatic measure and appreciation of results.

Recorded results of quality detection and capacity of generalization demonstrate the superiority performances given by the neural networks trained with Zernike moments feature vectors. Good localization rates, up to 94%, were achieved and accurate contours adapted to the faces shapes were obtained.

These results demonstrate also the high sensitivity of neural networks trained with Eigenfaces to the training database. A difference about 23% was recorded for them while only 7% of difference for those trained with Zernike moments.

We also found that for the two kinds of training feature vectors, best results were obtained for vectors sizes between 6 and 10. For sizes apart from this interval the quality of localization decreases considerably. In the case of sizes bellow 6, this will be due to insufficient information brought by vector parameters. For those larger than 10, neural networks became more complex and convergence more difficult.

Method performances can be improved by judicious choices on the training database size and contents and also by adapted parameters of training vectors and neural network. This method can be extended to face components detection and object detection in general.

## Acknowledgment

M. SAAIDIA is supported by a national grant from the ministry MESRS of the Algerian state.

## References :

- [1]. E. Hjelm et B. K. Low. "Face detection : A survey", Computer Vision and Image Understanding, vol. 83, no. 3, pp. 236-274, 2001
- [2]. M. Saaidia, S. Lelandais, V. Vigneron and M. Bedda, "Face detection using Zernike moments. Evaluation by quantitative measurement criterion", WSEAS Transactions on Signal Processing., Issue 2, Vol. 3, pp 226-232, 2007

- [3]. M.R. Teague, "image analysis via the general theory of moments", J. Opt. Soc. Amer., vol. 70, pp. 920-930, 1980
- [4]. M. Turk and A. Pentland, "Eigenfaces for recognition", Journal of cognitive Neuroscience, vol. 3, pp. 71-86, 1991
- [5]. J. Haddadnia, M. Ahmadi, and K. Faez, "An Efficient Feature Extraction Method with Pseudo Zernike Moment in RBF Neural Network Based Human Face Recognition System", EURASIP JASP, vol. 9, pp. 890-891, 2003
- [6]. S. Alirezaee, M. Ahmadi, H. Aghaeinia, and K. Faez, "A weighted pseudo-zernike feature extractor for face recognition", IEEE, ICSMC, vol. 3, pp. 2128-2132, 2005
- [7]. C. Rosenberger, A. Rakotomamonjy, B. Emile, "Generic target recognition", EUSIPCO vienne, pp. 1613-1616, 2004
- [8]. Y.L.Y. Zhuejie, "Face recognition with Eigenfaces", Proc. IEEE Intl. Conf. Industrial Technologies, pp. 434-438, 1994
- [9]. M. Rizon, M.H. Firdus, P. Saad, S. Yaacob, M.R. Mamat, A.Y.M. Shakkaf, A.R. Saad, Hdesa and M. Karthigayan, " Face recognition using Eigenfaces and Neural Networks", American Journal of Applied Sciences, vol. 2, pp. 1872-1875, 2006
- [10]. J. Gu, H. Z. Shua, C. Toumoulinb, and L. M. Luo, "A novel algorithm for fast computation of Zernike moments", Pattern Recognition, vol 35, pp. 2905-2911, 2002
- [11]. G. Amayeh, A. Erol, G. Bebis, and M. Nicolescu, "Accurate and efficient computation of high order zernike moments", First ISVC, Lake Tahoe, NV, USA, pp. 462-469, 2005
- [12]. K. Messer, J. Matas, J. Kittler, J. Luetin, and G. Maitre, "XM2VTSDB: The Extended M2VTS Database", In Second International Conference on Audio and Video-based Biometric Person Authentication, March 1999