Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 9, September 2013, pg.48 – 54

RESEARCH ARTICLE

FACIAL IDENTIFICATION SYSTEM USING MATRIX SPACE

Saurabh Mitra[1], Dr. Moushmi Kar[2], Shikha Mishra[3]
Saurabh.mit1000@gmail.com
[1,3]Faculty of Engineering, Dr. C.V. Raman University, Bilaspur, India

Abstract— In this research, we show how the promising Independent Component Analysis (ICA) technique extracts features that are more closely related to our intuition of discriminate information, and that improve the success rate compared to an equivalent system using PCA. The aim of the research is to implement ICA based face authentication system by extracting the invariant features of a face.

I. INTRODUCTION

Face authentication has gained considerable attention in the recent years, through the increasing need for access verification systems using several modalities (voice, face image, fingerprints, pin codes, etc.).Face authentication is different from face recognition (or classification): in authentication tasks, the system knows apriority the identity of the user (for example through its pin code), and has to verify this identity. In face authentication, as in most image processing problems, features are extracted from the images before processing [5].

Working with rough images is not efficient in face authentication, several images of a single person may be dramatically different, because of changes in viewpoint, in color and illumination, or simply because the person's face looks different from day to day. Therefore extracting relevant features, or discriminate ones, is a must. Nevertheless, one hardly knows in advance which possible features will be discriminating or not. For this reason, one of the methods often used to extract features in face authentication is Principal Component Analysis (PCA).

A. Face Authentication System

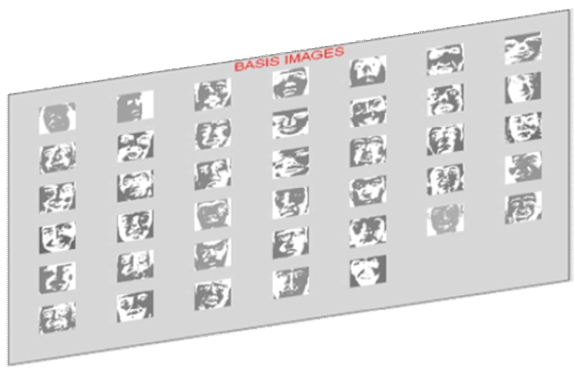
Face authentication system uses MATLAB as a platform where full face is taken as an input, and the invariant features are extracted and stored in the database for further processing using PCA and ICA. The ICA is using PCA as a pre- processing step.

In this module it just compares extracted features of the test image with the image of the same individual stored in the database. The image in the database is identified by ID No. The matching may be made in different ways, one being to take the Euclidean distance between vectors (extracted features). If the distance between the two vectors is lower than a threshold, it is considered to be a match otherwise rejected [3].

Face authentication systems typically compare a feature vector X extracted from the face image to verify with a client template, consisting in similar feature vectors Yi extracted from images of the claimed person stored in a database ($1 \leq i \leq n$, where n is the number of images of this person in the learning set). The matching may be made in different ways, one being to take the Euclidean distance between vectors. If the distance between X and Yi is lower than a threshold, the face from which X is extracted will be deemed to correspond with the face from which Yi is extracted. Choosing the best threshold is an important part of the problem: a too small threshold will lead to a high False Rejection Rate (FRR), while a too high one will lead to a high False Acceptance Rate (FAR); FRR and FAR are defined as the proportion of feature vectors extracted from images in a validation set being wrongly classified, respectively wrongly authenticated and wrongly rejected.

The validation and test sets must be independent (though with faces of the same people) from the learning set, in order to get objective results [2]. One way of setting the threshold is to choose the one leading to equal FRR and FAR. If the a priori probabilities of having false acceptances (impostors) and false rejections are equal, this corresponds to minimizing the number of wrong decisions, as a result of Bayes' law, other criteria could be considered, such as using individual thresholds for each person in the database. The human face image appearance has potentially very large intra-subject variations due to following:

i). 3D head pose
ii). Illumination (including indoor /outdoor)
iii). Facial expression
iv). Occlusion due to other objects or accessories
v). Facial hair
vi). Aging



B. Use Methods

ICA: ICA is a statistical method for transforming an observed multidimensional random vector into its components that are statistically as independent from each other as possible [10]. ICA is a special case of redundancy reduction technique and it represents the data in terms of statistically independent variables. ICA of a random vector consists of searching for a linear transformation that minimizes the statistical dependence between its components. The goal of ICA is to provide independent image decomposition and representation. ICA method can be distinguished from other methods since it looks for components that are both statistically independent and non-gaussian.

Infomax Algorithm: The goal in this algorithm is to maximize the mutual information between the environment X and the output of the neural network Y. The basic algorithm is as shown in Figure 1.2.The following steps is carried out:

(a)The R matrix (output of PCA function) which contains the sorted PCA coefficients is given as input [7]. This matrix is reduced to the size of no of principal components required. This in turn will also decide the no of independent components (ICs) for e.g. If we have 1000 images in the data base and would like to work with 200 principal components then x is given by x = R (1000:200).

Funct *W= gradBatchIca(X,λ,nbSweeps,batchSize)*

X = randomPermutation(X) / scramble the input vectors to improve stationarity */*

for i:= 1 to nbSweeps do

for j:= 1 to nbSamples step batchSize do

$$\Delta W := \sum_{k=0}^{batch\,Size} update(x_{j+k}, W)$$

$$W := W + \lambda\, \Delta W$$

end

end

Fig. 2: The basic INFOMAX Algorithm

Whitening: We transform the observed vector x linearly so that we obtain a new vector $\tilde{x}$ which is white (remove first and second order statistics from x), i.e. its components are uncorrelated and their variances equal to unity. In other words, the covariance matrix of $\tilde{x}$ equals the identity matrix:

$$E\{\tilde{x}\tilde{x}^T\} = I \qquad (1)$$

Learning: Learning is carried out through permuted $\tilde{x}$ that are of length M, in batch blocks of size B, adjusting weights, w, at the end of each block. The process is repeated every F counts till convergence. The learning rate, L may be decided by the system managers.

The different combinations of B and L will yield different results. For large numbers of rows in $\tilde{x}$ (e.g., 200), you need to use a low learning rate (e.g., 0.0005). Reduce if the output blows up and becomes Not a No (NAN). If lesser rows, then 0.001 or larger value should be used. We have used annealing learning rate, L, from 0.005 downwards to 0.001 towards end.

The updated weight matrix, w which was obtained by learning process is applied along with the whitening matrix, wz in the following equation:

Uu = w*wz*xx; (2)

Where xx holds original data, whitened mean extracted.

Uu gives the separated output signals.

F = uu'; Now, each row of F contains the ICA coefficients of one image.

Representations of Test Image: The pre-processed test image is contained in the row of Ctest. The centring of test image is carried out with respect to mean of training images using the following equation:

Dtest = Ctest-ones (1, 1)*mean(C) (3)

The PCA coefficients of test image are obtained:

Rtest = Dtest*V (4)

Where Rtest gives the PCA representations of test image, V is the Eigen vectors of training images.

The ICA representations of the test image are obtained similar to the training images in Ftest which is given by:

Ftest = w * wz * zero mean (Rtest (:1:200)')' (5)

Where, zero mean function gives the whitened mean extracted if we are taking 200 principal components.

Cosine Distance:

$$C = \frac{B_{test} . B_{train}}{\| B_{test} \| \| B_{train} \|} \qquad (6)$$

It computes the cosine (normalized dot product) between training vectors and test vector [6]. Output is a matrix of cosines. A cosine similarity measure is equivalent to length- normalizing the vectors prior to measuring Euclidean distance when doing nearest neighbor. The shortest distance gives the nearest match for the test image from the training images. Refer figure for cosine distance bar graph [8].

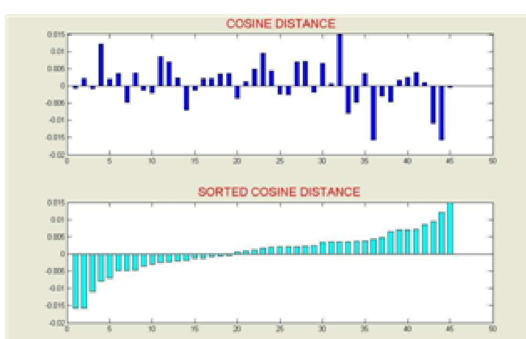


Fig. 5: Cosine distances of test image from the training images

Fig. 5 shows the Cosine distances of test image from the training images: (a) Note that it shows an exact match for image in the database, and (b) The sorted Cosine distances are shown.

Setting up of Threshold value and showing up of Result: The threshold value has been set by trial and error method. Presently it has been set to 0.0140.Where t1 is the minimum distance between the projected and the test image. If t1 < = 0.0140, then the face is identified else the system shows the nearest probable matches. Face Authentication is carried out for 1:1 matching as compared to 1: N in face recognition where N is the No of trained images in the database.

Image Search: The face database is searched for the ID No corresponding to the test image [9]. The pre-processed image corresponding to the ID No is selected for further procession

C. Architectures for Performing ICA on Images

Let X be a data matrix with n_R rows and n_c columns. We can think of each column of X as outcomes (independent trials) of a random experiment. We think of the row of as the specific value taken by a random variable across independent trials. This defines an empirical probability distribution for X1, …, Xn in which each column of X is given probability mass 1/nc . Independence is then defined with respect to such a distribution. For example, we say that rows i and j of are independent if it is not possible to predict the values taken by Xj across columns from the corresponding values taken by Xi, i.e.,

$$P\ (Xi=u,\ Xj=v) = P\ (Xi=u)\ P\ (Xj=v) \text{ for all } u,V€R,$$

where P is the empirical distribution.

We have to find a good set of basis images to represent a database of faces. We organize each image in the database as a long vector with as many dimensions as number of pixels in the image.

There are at least two ways in which ICA can be applied to this problem: (a) we can organize our database into a matrix X where each row vector is a different image. This approach is illustrated in (Fig. 6). In this approach, images are random variables and pixels are trials. In this approach, it makes sense to talk about independence of images or functions of images. Two images i and j are independent if when moving across pixels, it is not possible to predict the value taken by the pixel on image j based on the value taken by the same pixel on image I; and (b) we can transpose X and organize our data so that images are in the columns of X. This approach is illustrated in (Fig. 6).

In this approach, pixels are random variables and images are trials. Here, it makes sense to talk about independence of pixels or functions of pixels. For example, pixel i and j would be independent if when moving across the entire set of images it is not possible to predict the value taken by pixel i based on the corresponding value taken by pixel j on the same image.

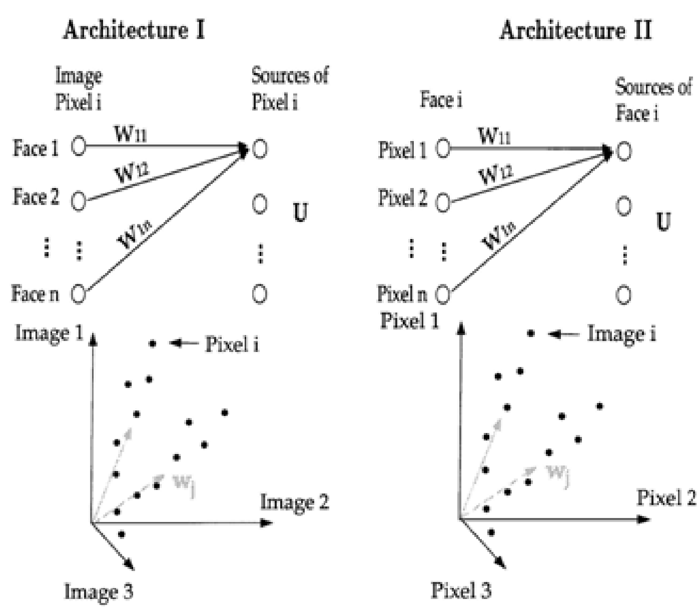


Fig. 6: Two architectures for performing ICA on images

(1) Architecture I for finding statistically independent basis images. Performing source separation on the face images produced IC images in the rows of U.

(2) The gray values at pixel location i are plotted for each face image. ICA in architecture I find weight vectors in the directions of statistical dependencies among the pixel locations.

(3) Architecture II for finding a factorial code. Performing source separation on the pixels produced a factorial code in the columns of the output matrix, U.

(4) Each face image is plotted according to the gray values taken on at each pixel location. ICA in architecture II finds weight vectors in the directions of statistical dependencies among the face images.

II. A Factorial Face Code

The goal in Architecture I was to use ICA to find a set of spatially independent basis images. Although the basis images obtained in that architecture are approximately independent, the coefficients that code each face are not necessarily independent. It is more robust to partial occlusion and local distortion Architecture II uses ICA to find a representation, in which the coefficients used to code images are statistically independent, i.e., a factorial face code. This display global properties i.e. they assign weights to potentially all pixels. It does not display local characteristics i.e. pixel not in local salient feature region still have nonzero values. Architecture II is more robust for face recognition as compared to Architecture I.

Therefore we have chosen Architecture II with Infomax Algorithm for our project. To achieve this goal, we organize the data matrix X so that rows represent different pixels and columns represent different images. This corresponds to treating the columns of Δ as a set of basis images. The

ICA representations are in columns of U=WIX. Each column of U contains the coefficients of the basis images in A for reconstructing each image in X (figure). ICA attempts to make the outputs, as independent as possible. Hence, U is a factorial code for the face images. The representational code for test images is obtained by:

$$\text{WI Xtest} = \text{Utest} \qquad (7)$$

Where Xtest is the zero-mean6 matrix of test images, and WI is the weight matrix found by performing ICA on the training images.

The basis images were each associated with a set of independent "causes," given by a vector of coefficients in S. The basis images were estimated by Δ, where W is the learned ICA weight matrix.

$$\text{ICA factorial representation} = (u_1, u_2, \dots, u_n)$$

The factorial code representation consisted of the independent Coefficients, u, for the linear combination of basis images in A that comprise each face image x.

In order to reduce the dimensionality of the input, instead of performing ICA directly on the 50 image pixels, ICA was performed on the first 10 PCA coefficients of the face images.

The Architecture II representation for the training images is contained in the columns of U, where $W_I R^T 10$ =U. The ICA weight matrix WI was 10x10, resulting in 10 coefficients U in for each face image, consisting of the outputs of each of the

ICA filters. The architecture II representation for test images was obtained in the columns of Utest as follows:

$$W_I \; R_{test}^{T} = U_{\;test}$$

The basis images for this representation consisted of the columns of Δ. A sample of the basis images is shown in Fig. 7, where the PC reconstruction P_{10}^{A} was used to visualize them. In this approach, each column of the mixing matrix W-1 found by ICA attempts to get close to a cluster of images that look similar across pixels. Thus, this approach tends to generate basis images that look more face-like than the basis images generated by PCA, in that the bases found by ICA will average only images that look alike. Unlike the ICA output U, the algorithm does not force the columns of A to be either sparse or independent. Face recognition was carried out for the coefficient vectors by the nearest neighbor algorithm, using cosines as the similarity measure.

$$C = \frac{B_{test} \cdot B_{train}}{\| B_{test} \| \| B_{train} \|} \qquad (8)$$

This equation computes the cosine (normalised dot product) between training vectors and test vector. An output is a matrix of cosines. A cosine similarity measure is equivalent to length-normalizing the vectors prior to measuring Euclidean distance when doing nearest neighbor. The shortest distance gives the nearest match for the test image from the training images.

III. RESULTS

Face authentication experiments were carried out with dataset I.

TABLE 1: AUTHENTICATION RATES

S. No	THRESHOLD		
	0.0140	0.0150	0.0160
Exactly same	100%	100%	100%
Slightly varying	70.21%	81.94%	96.74%

Experiments were carried out on dataset I. It was found that the variation in the recognition rate was not much when the no. of principal components were increased it shows with trials that maximum amount of information lays in the first 10% of the principal components. The results are also in consonance with the first experiment carried out on dataset. In the third experiment the variation in threshold was carried out for face authentication.

It was found that as the threshold was increased false acceptance rate (FAR) also increased. On the other hand false rejection rate (FRR) increased when threshold was reduced. It was also noticed that threshold was different for data based on different Asian (w Thehitish complexion) and Europeans (Fair complexion) [1].Thus system manager can decide the threshold depending on the type of requirements of the organisation. It also emerged from above experiments that as the database increase FRR decreases. Refer the graphs for Euclidean distance and cosine distance (Fig. 7 and Fig. 8). The graph gives the distances of different images from the test image [4]. It has been noticed that images of individuals which is quite common face is generally is part of first five probable matches of all the many test images.

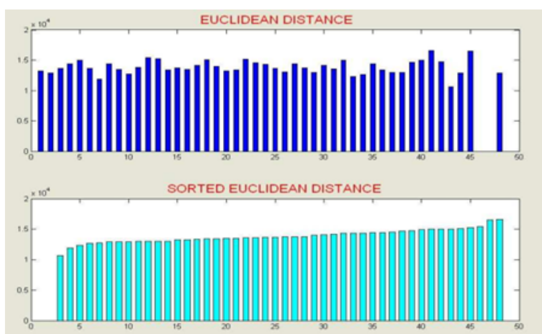


Fig. 7: The Euclidean distances of test image from the training images

Note that Fig. 7 shows an exact match for 46th and 47th image in the database, and the sorted Euclidean distances are also shown.

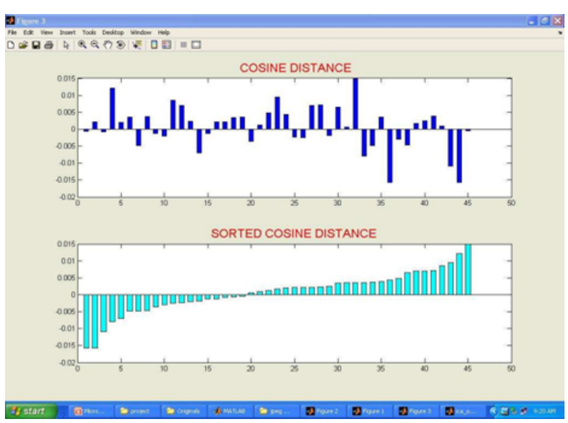


Fig. 8: The Cosine distances of test image from the training images

Note that Fig 8 shows an exact match for image in the database, and the sorted Cosine distances are also shown.

IV. CONCLUSIONS AND FUTURE WORK

ICA representations are designed to maximize information transmission in the presence of noise and, thus, they may be more robust to variations such as lighting conditions, changes in hair, make-up, and facial expression, which can be considered forms of noise with respect to the main source of information in our face database: the person's identity.

Future work can use the programs generated in this project work to further analyse and quantify the sensitivity of these programs to parameters such as learning rate and block size. The biometrics has always been a matter of curiosity for researchers.

REFERENCES

[1] Farhan Zaidi, T.Morris, P.Blenkhorn, "Blink detection for real time eye tracking", Journal of Network and Computer Applications (2002) pages 129-143 February 2002.

[2] Michael Chau and Margrit Betke "Real time eye tracking and blink detection using USB camera", Boston University Computer Science Technical Report No. 2005-12.

[3] D. G. Evans, R. Drew, P. Blenkhorn, "Controlling Mouse Pointer Position Using an Infrared Head Operated Joystick", IEEE TRANSACTIONS ON REHABILITATION ENGINEERING, VOL. 8, pages107-116, MARCH 2000.

[4] Y.I. Tian, T. Kanade & J. Cohn 2000. Dual-State Parametric Eye tracking. Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, pages 26-30 March 2000.

[5] J.Y. Deng & F. Lai. "Region-Based Template Deformation and Masking for Eye-Feature Extraction and Description". Pattern Recognition 30, pages 403-419, MARCH 1997.

[6] C. Fagiani, M. Betke, and J. Gips. Evaluation of tracking methods for human-computer interaction. Proceedings of the IEEE Workshop on Applications in Computer Vision (WACV 2002), pages 121–126, Orlando,Florida, December 2002.

[7] Y. Tian, T. Kanade & J. Cohn 1999. "Multi-State Based Facial Feature Tracking and Detection.Robotics Institute", Carnegie Mellon University, Technical Report CMU-RI-TR-99-18.

[8] X. Xie, R. Sudhakar & H. Zhuang. On Improving Eye Feature Extraction Using Deformable Templates. Pattern Recognition 27, pages 791-799,1994.

[9] A. L. Yuille, D. S. Cohen & P. W. Hallinan. Feature Extraction from Faces Using Deformable Templates. Proceedings Computer Vision and Pattern Recognition, pages 104-109, 2008.

[10] X. Wei, Z. Zhu, L. Yin, and Q. Ji. A real-time face tracking and animation system. Proceedings of the CVPR Workshop on Face Processing in Video (FPIV2004), Washington, D.C., June 28 2004.