

# Face Identification by Real-Time Connectionist System

Pedro Galdámez and Angélica González

University of Salamanca, Plaza de los Cados, 37008 Salamanca  
{peter.galdamez, angelica}@usal.es

**Abstract.** This document provides an approach to biometrics analysis which consists in the location and identification of faces in real time, making the concept a safe alternative to Web sites based on the paradigm of user and password. Numerous techniques are available to implement face recognition including the principal component analysis (PCA), neural networks, and geometric approach to the problem considering the shapes of the face representing a collection of values. The study and application of these processes originated the development of a security architecture supported by the comparison of images captured from a webcam using methodology of PCA, and the Hausdorff algorithm of distance as similarity measures between a general model of the registered user and the objects (faces) stored in the database, the result is a web authentication system with main emphasis on efficiency and application of neural networks.

**Keywords:** Neural networks, eigenfaces, Hausdorff distance, Face Recognition.

## 1 Introduction

In social relationships our principal target is the face, playing an important role in the perception of identity and emotions. Although the ability to relate intelligence or character from facial appearance is mistaken, the human ability to recognize faces is remarkable. Humans are able to recognize thousands of faces learned throughout their life and identify familiar faces at a glance. It is an amazing ability, pondering the great changes in the visual stimulus, given the conditions of observation, expression, aging, and distractions such as glasses, or changes in the hair style.

Face recognition has become a major issue in several applications, ranging from security systems, credit card verification to criminal identification; where the ability to modeling a face and distinguish it from a large number of stored models greatly improve criminal identification processes. During the 80s facial recognition work was inert. And later on in the 90s, a significant growth and interest in the investigation was started. Although there are different approaches to the problem of face recognition, there are two basic methods of which most approaches build their base. The first one is based on the concepts of information theory, that means, in the

methods of principal component analysis. In this approach, the most relevant information that best describes a face image is derived from the entire face. Based on the Karhunen Loeve expansion in pattern recognition M. Kirby and L. Sirovich Have shown [5] that any particular face could be mathematically represented in terms of a coordinate system that is called “eigenfaces”. The second method is based on extracting feature vectors of the basic parts of a face such as eyes, nose, mouth and chin. In this method, with the help of deformable templates and math, key information of the basic parts of a face meets and then becomes a feature vector [14].

## 2 Facial Recognition Using Eigenfaces

The face recognition algorithm with eigenfaces is described basically in the figure 1. First, the original images of the training set are transformed into a set of eigenfaces  $E$ , Then, weights are calculated for each image on the ( $E$ ) set, and then are stored in the ( $W$ ) set. Observing an image  $X$  unknown, weights are calculated for that particular image, and stored in the vector  $W_X$ . Subsequently,  $W_X$  compared to the weights of images, which is known for sure that they are faces (the weights of the training set  $W$ ).

### 2.1 Classification of a New Face

The process of classifying a new face in the  $\Gamma_{new}$  to another category (known faces) is the result of two steps. First of all, the new image is transformed into its eigenface components. The resulting weights forms the weight vector  $\Omega_{new}^T$ .

$$\omega_k = u_k^T (\Gamma_{new} - \Psi) \quad k = 1, \dots, M'$$

$$\Omega_{new}^T = [\omega_1 \ \omega_2 \ \dots \ \omega_{M'}]$$
(1)



**Fig. 1** Facial recognition algorithm based on eigenfaces

The Euclidean distance between two vectors of weights  $d(\Omega_i, \Omega_j)$  provides a measure of similarity between the corresponding images  $i$  and  $j$ . If the Euclidean distance between  $\Gamma_{new}$  and the rest of images on average exceeds a certain threshold value, through this can be assumed that  $\Gamma_{new}$  is not a recognizable face [8].

### 2.2 The Core System

The previous section discussed the procedure for obtaining the eigen vectors of a collection of images, in order to perform the classification. The flow of the algorithm based on this technique is explained below. The following method provides a comprehensive procedure applied, it describes the methods of preprocessing and flow catches ranging from the acquisition of the images to their identification, which subsequently will make emphasis in the latest techniques applied in the heart of the system to achieve the goal of face recognition.

#### Data Acquisition

The first activity to do is to take pictures of users to recognize them later. The project uses a general purpose webcam, this represents the first contact with the users logon screen, where he will decide whether or not to register a user. If so, when trying to login, the system will identify him. If the user is not recognized by the system, he or she can be registered on a screen dedicated for that purpose. Captured images are processed and stored in a database as an array of bytes, the original image is stored along with the grayscale image and the extracted face is capture.

#### Image Processing

Once users have been registered in the system, it is possible to make identifications, for which we proceed to capture the image of the login screen, the image is resized

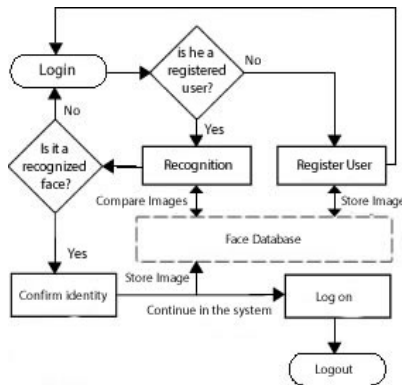


Fig. 2 Basic flow system

to  $320 \times 240$  pixels. The captured image is converted to grayscale, in this step the face is extracted and adjusts its size to  $100 \times 100$  pixels. Both actions use the library EmguCV [3]. For better results, the method needs to remove the background applying a mask that is just a black ellipse around the identified face, finally the procedure equalize the image, standardizing their lighting. The process is performed for all images in the database and for the image captured at the time of the login attempt.

### User Recognition

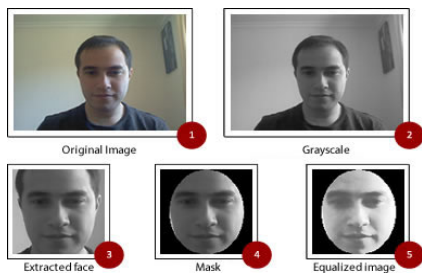
With all the pictures processed, an array is created with labels indicating to whom belongs each image, the eigenfaces and eigenvalues are calculated with the PCA algorithm. Subsequently, it obtains the Euclidean distance of the comparison of the weight vectors obtained. Defining a threshold value of 1000 representing a similarity between two images of 90%, under the assumption that a smaller distance means a greater similarity between the sets.

In other words, the eigenvectors are obtained from the set of images stored in the database, the eigenfaces and unknown image are compared both with the distance measurement, resulting in a distance vector from the input image regarding the collection of the database. The lowest value is obtained and compared with the threshold, being the label with the lower value the identified user.

## 3 Neural Network for Face Recognition

Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, speech, vision and control systems. In this project, there is a neural network that identifies each person in the database. After calculating the eigenfaces, the feature vectors of the faces are stored in the database. These vectors are used as inputs to train the network. In the training algorithm, the vectors of values belonging to a person, are used as positive for returning said individual neuron 1 as the neuron output assigned to that user and 0 in other neurons.

When the new image has been captured, the feature vectors are calculated from the eigenfaces obtained before, we compute new descriptors of the unknown image.



**Fig. 3** Processing Flow of an Image

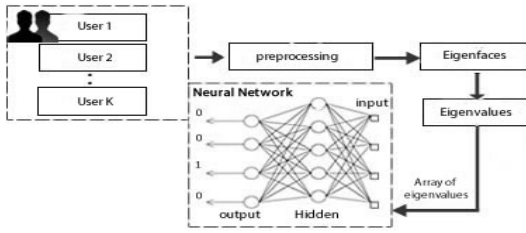


Fig. 4 Neural Network Training

These descriptors are entered into the neural network, the outputs of individual neurons are compared, and if the maximum output level exceeds the predefined threshold, then it is determined that the user belongs to the face assigned to the neuron with the index activated.

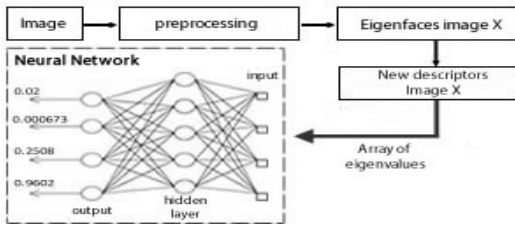


Fig. 5 Neural Network Simulation

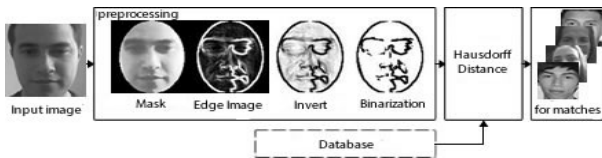
### 3.1 Algorithm Implementation Summary

The approach of face recognition using eigenfaces and neural networks can be summarized in the following steps:

1. Build a library (database) of faces.
2. Choose a training set ( $M$ ) that includes a number of images for each person with a variation in expression and lighting.
3. Construct the matrix  $L$  of  $M \times M$ , find its eigenvalues and eigenvectors, and choose the vectors  $M'$  with the highest values.
4. For each image in the database, calculate and store its feature vector.
5. Create a neural network with one output neuron per registered user.
6. Train the network, using the images of each user as positive examples of neurons and negative examples for the rest.
7. For each new image calculate its eigenvector and get their eigenvalues.
8. Use the new vector calculated as input to the network.
9. Select the neuron with the maximum value. If the output of the selected neuron passes a predefined threshold, is presented as the recognized face.

## 4 Application of the Hausdorff Distance

The Hausdorff distance measure used in this document is based on the assumption that the facial regions have different degrees of importance, where characteristics such as eyes, mouth, face contour and others; play the most important role in face recognition. The algorithm applied is based on what is stated in [6]. In applying the Hausdorff distance, operates basically the comparison of edge maps. The advantage of using edges to match two objects, is that this representation is robust to illumination change. Accordingly, the edge detection algorithm used will have a significant effect on performance.



**Fig. 6** Flow applying the Hausdorff distance

The procedure involves removing the background of the image as it was performed in the preprocessing original, added some steps after image masking, we proceed to obtain the edges using the Sobel filter, the image is reversed to operate with a white background, then the face is binarized, similar procedure is applied to each image stored in the database.

With the obtained objects, as if they were geometric figures performing a comparison process, calculating the Hausdorff distance, we compare pixels to get how similar are the two figures, resulting in a collection of values that contain the distance of the input image with respect to each item in the database. The object having the smaller relative distance can be presented as an option, if not exceeds the minimum threshold value, identifies the user, otherwise the problem is considered as an unsolved. In the developed system, the Hausdorff algorithm is presented as an alternative to the neural network and recognition using eigenfaces, if the three procedures identify that the user is the same, even without exceeding the thresholds defined in each process, the image is accepted belongs to user input identified by all three techniques.

## 5 Results and Experimental Set-Up

The proposed method has been tested on 10 different users. Each user has more than one image with different conditions (expression, lighting). For the PCA algorithm is defined a threshold of 90%. Regarding neural network the number of output neurons used is equal to the number of people in the database, the parameters are defined:

- Type: FeedForward, resilient backpropagation.
- Number of layers: 3 (input, hidden layer, output).
- Input layer neurons:  $M$  (number of eigen values obtained to describe faces).
- Neurons in the hidden layer: 10
- Neurons in the output layer: one per registered user in the database.
- Recognition threshold: 90%

With respect to the algorithm Hausdorff, we defined a threshold value of 90%, the result is conditioned by the response of the neural network and PCA algorithm. Only the user shall be deemed recognized by the Hausdorff distance if the three procedures have agreed in their response. In the experiment, we developed a Web application using EmguCV library, that is a wrapper for OpenCV to implement the PCA algorithm and face detection using ViolaJones cascade classifier. We decided to replace the traditional security mechanism of passwords by using HTML5 canvas object that was received from a video which it was generated by a web cam with the facial recognition information. When the algorithm was implemented, we got a number of false positives the result was about two in ten attempts for the login process, this situation could increase to three if is modified the worst lighting conditions. By including the neural network, sending as input the calculated Eigen vectors, the recognition process was improve in nine out of ten attempts. Finally using the Hausdorff distance was observe that the system became robust to different background conditions and lighting changes, but even with that algorithm the face recognition was not perfect and we got nine of ten positives results in face recognition process.

## 6 Conclusions and Future Work

In this research, we studied three approaches for face recognition. The first based on an approximation Eigenfaces using Euclidean distance for comparison of results, the second approach use the eigenvalues of images as input to a neural network that performs recognition and finally we applied another measure, the Hausdorff distance on edges maps of images of registered users. The main advantage of working in this format is the independence over the lighting problem. The result is a web site with a login using the user's face. The PCA technique allows to obtain the most important components of a face, its strength on feature based approaches is its simplicity and speed, that combined with the application of neural networks provides a learning potential acceptable for use in web applications. This methodology can be extended by replacing the Euclidean distance for a more robust approach to consider the image obtained, and distinguish between sections within it, assigning weights. Finally in a system that is robust and secure, you should consider the changes in lighting, the presence of details in the face such as glasses, beard, and more variables considering the background of the image. Project Experimental results show that the orientation of the face and background of image, represent a major drawback, as future work, the system could try detecting rotation angle of the face.

## References

1. Goldstein, A.J., Harmon, L.D., Lesk, A.B.: Identification of human faces. *Proc. IEEE* 59, 748–760 (1971)
2. Intel, Intel Open Source Computer Vision Library, v2.4.1 (2006), <http://sourceforge.net/projects/opencvlibrary/>
3. Intel, EmguCV Envoltorio de la biblioteca OpenCV, v2.4.0 (2012), <http://www.emgu.com>
4. Kerin, M.A., Stonham, T.J.: Face recognition using a digital neural network with self-organizing capabilities. In: *Proc. 10th Conf. on Pattern Recognition* (1990)
5. Kirby, M., Sirovich, L.: Application of the Karhunen-Loeve procedure for the characterization of human faces. In: *IEEE PAMI*, vol. 12 (1990)
6. Lin, K.-H., Lam, K.-M., Siu, W.-C.: Spatially eigen-weighted Hausdorff distances for human face recognition. Polytechnic University, Hong Kong (2002)
7. Manjunath, B.S., Chellappa, R., Malsburg, C.: A feature based approach to face recognition. *Trans. of IEEE*, 373–378 (1992)
8. Dimitri, P.: Eigenface-based facial recognition (Diciembre 2002)
9. Rowley, H., Baluja, S., Kanade, T.: *Neural network face detection*, San Francisco, CA (1996)
10. Smith, L.I.: A tutorial on principal components analysis (February 2002), [http://www.cs.otago.ac.nz/cosc453/student\\_tutorials/principal\\_components.pdf](http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf) (accessed on April 27, 2012)
11. Terrillon, J., David, M., Akamatsu, S.: Automatic detection of human faces in natural scene images by use of a skin color model, Nara, Japan (1998)
12. Turk, M., Pentland, A.: Eigenfaces for recognition (1991a), <http://www.cs.ucsb.edu/mturk/Papers/jcn.pdf> (accessed on April 27, 2012)
13. Viola, P., Jones, M.J.: Robust real-time face detection. *International Journal of Computer Vision* (2004)
14. Yuille, A.L., Cohen, D.S., Hallinan, P.W.: Feature extraction from faces using deformable templates. In: *Proc. of CVPR* (1989)