A neural network framework for face recognition by elastic bunch graph matching

Francisco A. Pujol López, Higinio Mora Mora*, José A. Girona Selva

***Abstract*—** Automated biometric systems are being widely used in many applications. Face recognition is one of the most promising methods due to its good acceptance by users. In this paper, we have explored including neural networks models for face recognition combined with other methods. As a result, a new recognition algorithm based on Elastic Bunch Graph Matching and Self-Organizing Maps is introduced. In this algorithm a combination of global and local techniques are applied to construct graphs whose nodes encode the facial features. A formal framework for specifying the functions involved is defined. The experiments performed were aimed at calibrating the map and evaluating the performance. The experimental results show the effectiveness of the proposal when compared to other well-known methods.

***Keywords*—** Pattern recognition, Face Recognition, Neural Networks, Self-Organizing Maps.

I. Introduction

IN recent years, there has been an intensive research to develop complex security systems involving a new kind of *'key'*: the biometric features. Automated biometric systems are being widely used in many applications such as surveillance, digital libraries, forensic work, law enforcement, human computer intelligent interaction, and banking, among others. For applications requiring high levels of security, biometrics can be integrated with other authentication means such as smart cards and passwords. In relation to this, face recognition is an emerging research area and, in the next few years, it is supposed to be extensively used for automatic human recognition systems in many of the applications mentioned before.

One of the most popular methods for face recognition is the Elastic Graph Bunch Matching (EBGM), proposed in [1], and it is an evolution of the method known as Dynamic Link Architecture (DLA) [2]. The main idea in the elastic graph matching is to represent a face starting from a set of reference or fiducial points known as landmarks. These fiducial points have a spatial coherence, as these points are connected using a graph structure. Considering these nodes, geometric information can be extracted and both distance and angle metrics can be defined accordingly.

This algorithm takes into account that actual facial images have many nonlinear features (variations in lighting, pose and expression) that are not generally considered in linear analysis methods, such as LDA or PCA. Moreover, it is particularly robust when out-of-plane rotations appear. However, the main drawback of this method is that it requires of an accurate location of the fiducial points.

Artificial Neural Networks (ANN) is one of the most used paradigms to address problems in artificial intelligence. Among the techniques and architectures proposed by the scientific community in this field, the Self Organizing Map (SOM) has special features for association and pattern classification [3], and it is one of the most popular neural network models. The key aspect that characterizes the family of problems in which it is desirable to apply this technique is the inaccuracy or lack of formalization of problems. In these cases, there is not a precise mathematical formulation of the relationship of input patterns [4].

Although there have been many scientific advances in the field of artificial intelligence and facial recognition, it is still not clear enough how the human brain recognizes different faces. This reasoning motivates the application in this work of neural network techniques to the face recognition problem with the goal of improving existing approaches.

Consequently, in this paper we will use ANNs to improve the efficiency of the EBGM algorithm. To do this, a SOM is applied in the construction of the database of facial graphs in an adaptive learning process. First, the fiducial points will be extracted automatedly and, after that, known faces will be grouped (or clustered) into *M* classes, each class corresponding to a different person.

This paper is organized as follows: Section II describes the EBGM method and summarizes the related work in the domain of using that method for face recognition; Section III explains the proposal of an EBGM-based face recognition method and the formal framework to define it; Section IV introduces the neural network approach with a Self Organizing Map for recognition; Section V describes the experiments carried out; and finally, conclusions and some future work are discussed in Section VI.

* H. Mora-Mora, MT. Signes-Pont, J. Azorín-López and L. Corral-Sánchez are with the Department of Computer Technology and Computation, University of Alicante, Spain, 03690, San Vicente del Raspeig, Alicante, Spain. e-mail: ({fpujol, hmora, jags20}@ua.es).

II. EBGM Algorithm and Related Work

In this section, the EBGM algorithm is described and, afterwards, some recent, related works are discussed.

A. The Elastic Bunch Graph Matching

Elastic Bunch Graph Matching is a *feature-based* face identification method. EBGM derives a bunch of jets for each training image and uses the jets to represent the graph node. To form a bunch graph, a collection of facial images is marked with node locations at defined positions of the head. These node locations are called landmarks and are obtained by a semiautomatic process. When matching a bunch graph to an image, the jet extracted from the image is compared to all jets in the corresponding bunch attached to the bunch graph and the best matching one is selected.

Jets are defined as *Gabor coefficients* in a landmark location computed by convoluting a set of Gabor wavelet filters around each landmark location. The jets of all training images are collected in a data structure called a bunch graph. The bunch graph has a node for every landmark on the face and every node is a collection of jets for the corresponding landmark. The main steps for face recognition by EBGM are outlined below [5]:

- *Step 1*: Select the landmarks on the training face images to create the face models. The selection is performed manually.
- *Step 2*: Convolve these points with a Gabor wavelet to construct the Gabor jets *J*. The local appearance around a fiducial point $\vec{x}$ will be coded by using the convolution of the input image $I(\vec{x})$ with a Gabor filter $\psi_m(\vec{x})$, so that:

$$\mathrm{J}_j(\vec{x}) = \int \mathrm{I}(\vec{x}') y_j(\vec{x}-\vec{x}') d^2\vec{x}' \tag{1}$$

where

$$\psi_m(\vec{x}) = \frac{\left\| \overrightarrow{k_m} \right\|}{\sigma^2} \exp\left(-\frac{\left\| \overrightarrow{k_m} \right\|^2 \left\| \vec{x} \right\|^2}{2\sigma^2} \right) \cdot \left[\exp(i\, k_m \cdot \vec{x}) - \exp(-0.5\sigma^2) \right] \tag{2}$$

and $\overrightarrow{k_m}$ is the wave vector:

$$k_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v \cos\varphi_\mu \\ k_v \sin\varphi_\mu \end{pmatrix},\ k_v = 2^{-\frac{v+2}{2}}\pi,\ \varphi_\mu = \mu\frac{\pi}{8}. \tag{3}$$

Consequently, 5 different frequencies (v = 0, 1, …, 4) and 8 different orientations (μ = 0, 1, …, 7) are used and, as a result, a jet will have 40 coefficients.

- *Step 3*: Create a data structure called bunch graph corresponding to facial landmarks that contains a bunch of model jets extracted from the face model.
- *Step 4*: Then for every new image to be recognized:

(a) Estimate and locate the landmark positions with the use of the bunch graph. (b) Calculate the new jets displacement from the actual position by comparing it to the most similar model jet. (c) Create a new face graph containing each landmark position and jet values for that landmark position.

- *Step 5*: Similarly, for each new image, estimate and locate the landmarks using bunch graph. Then the features are extracted by convoluting with the number of instances of Gabor filters followed by the creation of face graph. The matching score is calculated on the basis of similarity between face graphs of images in the database and the one in a new input image.

B. Related work

EBGM has been used for recognition in the last few years. Most of the methods based on EBGM use Gabor wavelets for feature extraction [6]. These features are represented by a grid of points geometrically adjusted to the features extracted. The recognition is based on the wavelet coefficients, which are calculated for the nodes of a 2D elastic graph representing the grid containing the landmarks. This method combines a local and a global representation through the processing of a Gabor filter with several scales and several directions (jets), of a point set –called fiducial points– located in specific regions of the face. The location of the fiducial points is the most complex task of this method. These points depend on lighting conditions, the expression and the pose of the face.

An alternative method proposed in [7] is the application of the histogram of orientation gradients (Histogram of Oriented Gradients, HOG) instead of using Gabor filters to locate features. This algorithm provides invariance in terms of location and orientation. To do this, the reference points are extracted in the space representation and the gradients of the image with respect to the dominant orientation around each reference point are calculated. Finally, the HOG descriptor, which is a statistic measure where the orientations of all the image gradients around a reference point are taken into account, is calculated.

Recently, a combination of EBGM with PCA and soft biometrics is used to make a study on the influence of age variations in face recognition [8]. Additionally, some new versions of EBGM focus on fast versions of the algorithm in order to make it feasible for real conditions; thus, a parallel version of EBGM for fast face recognition using MPI (Message Passing Interface) is presented in [9]; the authors divide the training process into *p* processors, then the recognition process is made simultaneously. Khatun and Bhuiyan [10] presented a neural network based face recognition system using Gabor filter coefficients, where the recognition used a hybrid neural network with a two networks, a Bidirectional Associative Memory (BAM) for dimensional reduction of the feature matrix to make the recognition faster and a Multilayer Perceptron with backpropagation algorithm for training the network.

In [11] a data mining approach to improve the performance of EBGM in case of using a large database was proposed, based on an entropy decision tree with the most important features in the face recognition process. Sarkar [12] combines skin detection with EBGM so as to obtain an accurate recognition, since skin segmented images remove background noises and reduce errors in identifying Gabor features. Finally, Li and Wachs [13] applied EBGM to hand gesture recognition, where a hierarchy is assigned to each node of the graph and the classification of hand gestures is performed against complex backgrounds. The location of the fiducial points is one of the most complex tasks of the method. Their positions highly depend on the lighting conditions, facial expressions and pose. In the original EBGM algorithm, a fixed number of features were established. These features corresponded to specific face characteristics, such as the pupils or the corners of the mouth. As a result, a facial model graph is obtained and the fiducial points are manually selected for each image in the database.

Another way to locate the features is based on a uniformly distributed grid of points that deforms and conforms to a pattern, such as the contours identified by an edge detector (Canny, Sobel, MLSEC, etc.) [14, 15].

Some recent advances have been made for the detection of the fiducial points in faces. Thus, Belhumeur et al. [16] used a Bayesian model that combines the output of local detectors with a non-parametric set of global models for the part locations based on a set of hand-labeled face images. The experiments were performed both using the BioID database and the new Labeled Face Parts in the Wild (LFPW) database, with very accurate results in any case. The Labeled Faces in the Wild (LFW) database was used instead in [17], where a method based on regression forests that detects 2D facial feature points in real-time is presented.

Other recent relevant works include: Baltrusaitis et al. [18] proposed a probabilistic patch expert (landmark detector) that can learn non-linear and spatial relationships between the input pixels and the probability of a landmark being aligned. Then, the 2-D fiducial detection method proposed in EBGM is extended in [19] to independently detect fiducial points by restricting the search range corresponding to each target fiducial, thereby removing the computationally expensive iterative scheme present in the original EBGM. Finally, Jin et al. [20] developed a Hough voting-based method to improve the efficiency and accuracy of fiducial points localization.

To sum up, from this revision two conclusions emerge: first of all, there is still a great interest from many research groups in order to use and improve the original EBGM method for face recognition; moreover, most of these investigations are focused on adapting EBGM to be used in real-time conditions with an accurate location of the landmarks or fiducial points for faces. It is clear that there is still much work to do in this field, and no previous works on the application of neural networks to EBGM have been found.

III. A PROPOSAL OF AN EBGM-BASED FACE RECOGNITION METHOD

To facilitate the work of connectionist models, in this work an adaption of basic EBGM method described in the previously section is performed.

Therefore, the faces are represented using a facial graph that includes geometric and textural information. The facial graph is defined as a pair $\{V, A\}$, where V refers to the set of vertices or nodes and A to the set of edges. Each vertex corresponds to a fiducial point and encodes the corresponding vector of jets and its location, that is, $V_i = \{J_i, P_i(x,y)\}$. Each edge A_{ij} encodes information on the distance and angle between the two nodes it connects, so that $A_{ij} = \{d_{ij}, \theta_{ij}\}$.

For each node, a 2-dimensional histogram $hist_i$ is constructed. In this histogram, the information about the distance $D = \{d_{i1}, d_{i2}, ..., d_{in}\}$ and the angle $\theta = \{\theta_{i1}, \theta_{i2}, ..., \theta_{in}\}$ from node i to the other nodes in the graph will be stored. Therefore, the histogram $hist_i$ consist of k bins corresponding to x distance-intervals by y angle-intervals. Thus, the k bins in histogram $hist_i$ are uniformly constructed in a *log-polar* space. Each pair $(\log(d_{ij}), \theta_{ij})$ increases the corresponding histogram bin. The algorithm followed to obtain the fiducial points is represented by commented pseudo-code below:

Algorithm 1. Obtaining the fiducial points from face

```
1. Normalize image sizes.
2. Apply an edge detector. In this work, the well-known
   Canny edge detector [26] is used.
3. Create a grid of Nx x Ny points, where nodes are uniformly
   distributed.
4. Each node adjusts its position to the nearest point in the
   edges obtained in Step 2.
5. The distances and angles from each final node to the rest of
   nodes are calculated.
```

A Gabor jet J is now constructed. Following Wiskott's approach [1], a vector of 40 complex components will be obtained. A jet J is then obtained considering the magnitude parts only. The position of each of the nodes in both facial graphs is known, as each vertex V_i encodes this information: $V_1 = \{J_1, P\}$, $V_2 = \{J_2, Q\}$, where $P = \{p_1, p_2, ..., p_n\}$ and $Q = \{q_1, q_2, ..., q_n\}$ are the vectors with the positions of each of the fiducial points for both faces.

So that, just as basic EBGM method, in order to match two facial graphs, $G_1 = \{V_1, A_1\}$ and $G_2 = \{V_2, A_2\}$, both geometric and texture information will be used.

Three functions of similarity are proposed in this work: the *Match Cost Function* (MCF), the *Norm Vector Function* (NVF) and *Gabor Feature Match Function* (GFMF).

Taking into account the histograms previously computed with geometric information of the nodes, MCF is calculated adding the matching costs for each node in the input facial graph G_1 with its corresponding node in the stored facial graph G_2:

$$\mathrm{MCF}(G_1, G_2) = \mathrm{MCF}(P, Q) = \frac{\displaystyle\sum_{i=1}^{n}\sum_{k}\frac{[hist_{p_i}(k) - hist_{q_i}(k)]^2}{h_{p_i}(k) + h_{qi}(k)}}{\|P\| \cdot \|Q\|} \quad (4)$$

where $||P||$, $||Q||$ refer to the norm of vectors P and Q.

The NVF is calculated by adding the norm of the vector of differences among the matched nodes:

$$\mathrm{NVF}(G_1, G_2) = \mathrm{NVF}(P, Q) = \sum_{i=1}^{n}\left\|\overrightarrow{p_i c_P} - \overrightarrow{q_i c_Q}\right\| \quad (5)$$

where

$$c_P = \frac{1}{n}\sum_{i=1}^{n} p_i \text{ and } c_Q = \frac{1}{n}\sum_{i=1}^{n} q_i$$

The texture information given by the Gabor jets from each node will be used to define the third similarity function: *Gabor Feature Match Function* (GFMF); thus, for each node $pi \in P$, a jet J_{pi} is calculated. Let R contain the Gabor jets of all the nodes in a facial graph, $R = \{J_{p1}, J_{p2}, \ldots, J_{pn}\}$. The function GFMF between two facial graphs is calculated as follows:

$$\mathrm{GFMF}(G_1, G_2) = \mathrm{GFMF}(R_1, R_2) = \frac{1}{n}\sum_{i=1}^{n} < R_{1_i}, R_{2_i} > \quad (6)$$

where $< R_{1_i}, R_{2_i} >$ is the normalized dot product between the i-th jet in R_1 and the i-th jet in R_2 . As mentioned before, only the magnitude of the Gabor coefficients in the jets is considered.

Finally, from the expressions defined in (4) to (6), the final similarity function called *Global Distortion function* (GD) is defined, which combines the results from each of them:

$$\mathrm{GD} = \lambda_1\,\mathrm{MCF} + \lambda_2\mathrm{NVF} + \lambda_3\mathrm{GFMF} \quad (7)$$

where λ_1, λ_2 and λ_3 are coefficients to be obtained experimentally and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

The functions that make up GD are normalized to the range [0, 1]. This normalization is performed with the maximum values for each component function using the distance between the input graph facial and facial graphs stored in neurons. Finally, the GD function is as follows:

$$\mathrm{GD} = \frac{\lambda_1\,\mathrm{MCF}}{\max(\mathrm{MCF})} + \frac{\lambda_2\mathrm{NVF}}{\max(\mathrm{NVF})} + \frac{\lambda_3\mathrm{GFMF}}{\max(\mathrm{GFMF})} \quad (8)$$

The values of $\mathrm{GD} \in [0, 1]$, where facial images belonging to the same person will give results close to 0, and facial images of different people will have a high GD value. The correct acceptance threshold value will be calculated to perform a recognition rates according contextual application features.

IV. Improving EBGM Algorithm with a Neural Network Approach

A. Self Organizing Map formulation

The Self Organizing Map (SOM) is a Neural Network technique that implements a nonlinear projection from a high-dimensional space onto a low-dimensional array of neurons. That mapping tends to preserve the topological relationship of the inputs, so, the visual image of this map depicts clusters of input information and their neighbour relationships on the map [3], [21].

The utility of the Self Organizing Maps (SOMs) for recognition problems has been proven in numerous studies. Next, we describe the SOM formalization to the proposed facial identification process.

A SOM is defined at any time by a collection of neurons, their position on the map and the weight of each. The neurons are connected to adjacent neurons by a neighbourhood relation. This set up the topology or the structure, of the map. The topological configuration of neurons are generally rectangular or hexagonal grid [21]. For a SOM of M neurons, the set,

$$\mathrm{W}(\mathrm{t}) = \{\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_\mathrm{M}\} \quad (9)$$

has the weight information as a whole, where w_i is the weight vector associated to neuron i and is a vector of the same dimension of the input. The set W(t) evolves according the Self-Organizing Map algorithm. The neurons position, defined by their weight vector, are configuring a topological mapping of the input space.

Let $\mathrm{X} \in \mathbb{R}^\mathrm{k}$ be the input vector of the SOM. For application to face recognition, this vector consists of k features extracted from the face to identify. We define,

$$\Psi\text{: } ℽ \rightarrow \mathbb{R}^\mathrm{k} \quad (10)$$

as the function that obtains the characteristics of the face to make up the vector X. So that,

$$\forall\ \mathrm{I} \in ℽ,\ \Psi(\mathrm{I}) = \mathrm{X} \in \mathbb{R}^\mathrm{k} \quad (11)$$

A classification of the face images is obtained when running the SOM algorithm on feature vectors calculated with Ψ from images of faces. On this classification can be applied a clustering process according to a method known. There are numerous methods of clustering [22]. However, the representation clusters in a 2D region is usually not a simple problem because the input data is usually of a high dimensionality. Let,

$$\mathrm{G}_\Psi = \{\mathrm{g}_1,\ \mathrm{g}_2,\ \ldots,\ \mathrm{g}_\mathrm{m}\} \quad (12)$$

be the set of groups obtained when clustering process is made. So that,

$$\forall I \in \gamma \Rightarrow SOM(\Psi(I)) \in G_{\Psi} \quad (13)$$

where, SOM function is called to the classification function of a feature vector, from input of the map to one of the groups generated.

The working hypothesis of this research is to consider that the classification with SOM network, suitably trained with images of individuals (γ), is correct. That is, a bijective function between groups and individuals can be established.

The main research described in this paper is to determine the extent to which the above hypothesis is true and under what conditions. For this, SOM operation and feature extraction functions from a face are analyzed under various configurations.

In this way, aspects of the facial identification made through SOM classification network are defined, suitable feature extraction function Ψ is identified, and also, the characteristics of the similarity function and acceptance threshold will be established under the premise that this function should provide the minimum value for a face when it is classified within the cluster corresponding to its individual.

The above information will enable to calibrate the method of face recognition under different scenarios and conditions.

B. *Self Organizing Map configuration*

As mencioned previously, in this paper we propose a neural network as a function of face recognition. The self-organizing map is responsible for building itself the database of facial graphs from the training images. Specifically, a two-dimensional $N_x x N_y$ SOM neural network is used. The number of neurons of the map will be determined experimentally in order to establish the minimum size that maximizes the efficiency in identification.

To analyze and extract features from each image, let's use one of the techniques that provides better results. The EBGM method will obtain the data of each face to be used as inputs for the training phase of the network and subsequent recognition of the method. From these data, the SOM uses the features extracted as inputs. The identification threshold t consist of the maximum distance that characterizes the clusters organized into the SOM in the training process. In this case, SOM network applies a classification process from the set of face graphs obtained from training images and generates G_{Ψ} clusters where each of them corresponding to one of the individuals to be identified.

The input data to the SOM network come from the EBGM method output's face graphs. A face graph is the structure used to represent the face through EBGM method. This graph has the set of nodes corresponding to the set of landmarks of the face and, each of them contains both geometric and texture characteristics. The following figure shows the face graph representation as input matrix array.

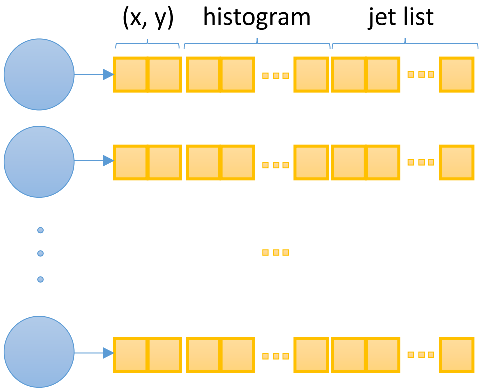

Fig. 1 SOM input data from facial graph

For practical implementation, the matrix arrangement of the entry represented in above figure can be transformed into one-dimensional vector placing continuously columns into memory.

The following algorithm shows the learning algorithm of the SOM.

Algorithm 2 Learning Algorithm Overview

```
1. Randomize the map's neurons' weight vectors.
2. Obtain face graph bunch using EBGM function of input face.
3. Every neuron is examined to calculate which one's weights are most like the input data.
4. The radius of the neighborhood of the winning neuron is now calculated.
5. The radius of the winning neuron is now updated according a neighborhood function.
6. Each neighboring node's (found in previous step) weights are adjusted to make them more like the input vector according a learning function.
7. Repeat step 2 for N iterations.
```

The calculation of the winning neuron are performed using for this similarity function defined in (13). Then, the neuron which minimizes the result of GD function is the winning neuron.

After this training process, there is a map composed of $N_x x N_y$ neurons organized according to the similarity between the input data. That is, the neurons which are near each other and are in the same cluster (g_i) have information about the face of the same user.

Once the self-organizing map has built and trained, it can be used as recognition function (F) to identify user by his/her face.

V. Results

A. *Experimental setup*

The recognition scheme has been tested with the FERET database [23]. The version used in this work is the Color FERET Database. It contains 11338 pictures of 994 different individuals. For our experiments we used the sets of images fa and fb, corresponding to 843 individuals with frontal images only. There is an image of each of the 843 individuals in fa and another one in fb. Images stored in fb were taken a few moments of time after the ones in fa, so in most cases some changes in the expression of the model can be noticed. All FERET images have a size of 512 x 768 pixels.

For the experiments, a set of 20 users were used, with 5 training images and 2 test images per user, with a resolution 128 x 192 pixels. The tests have been performed using Matlab® with a 2 GHz Intel Core i5 and 8 GB memory.

B. Experimental results

The experimentation made has a dual objective: in first place, it seeks to configure the SOM network to provide the best results, and secondly, it tests the effectiveness of the method in the process of facial recognition. The two types of experiments are combined in a set of tests on the input images. The following subsections focus on testing one aspect leaving the other aspects as invariant. In all cases, the number of training iterations made is 200.

1) Type and size of map

The following table shows the recognition rate, where N (20) is the number of different individuals used for the experimentation (and 5 different images for person).

TABLE I. RECOGNITION RATE (%): TYPE AND SIZE OF MAP

SOM size	rectangular grid (gridtop)	hexagonal grid (hextop)
N-10 x N-10	77.6	77.61
N-5 x N-5	80.9	80.9
N x N	88.6	89.7
N+5 x N+5	88.5	88.9
N+10 x N+10	87.3	88.5

The above table shows that the optimal size of the map is around the number of individuals to identify. As regard type, a hexagonal distribution of neurons provides slightly better results.

2) Size of facial graph

In this experiment, the connection between the size of facial graph obtained from EBGM method and recognition rate is analysed. The results are shown in the next table:

TABLE II. RECOGNITION RATE (%): SIZE OF FACIAL GRAPH

SOM size	Size of facial graph			
	6x6	8x8	10x10	12x12
N x N	86.1	87.4	89.6	89.7

It is observed that, at first seems, there are a correlation between the size of the graphs and the recognition accuracy. However, in this respect (as in the above), we must find a compromise between speed and accuracy of training, as training cost increases considerably with the size of the graph and the map.

3) Weighting λ coefficients of similarity function

The following table shows the results obtained according to various configurations of *λ*.

TABLE III. RECOGNITION RATE (%): SIZE OF FACIAL GRAPH

SOM size	λ coefficients weight		
	$\lambda_1=0.2, \lambda_2=0.7, \lambda_3=0.1.$	$\lambda_1=0.7, \lambda_2=0.2, \lambda_3=0.1.$	$\lambda_1=0.2, \lambda_2=0.2, \lambda_3=0.6.$
N x N	89.7	89.7	81.6

As seen in the above table, greater weighting metric GFMF not give good results, whereas the other two metrics shows that the best correct identifications is reached. Generally, it appears that giving high weight to the NVF and MCF functions the best results are obtained for experimentation.

C. Comparison with other methods

A set of experiments have been performed in order for our system to be compared with some other existing algorithms for face recognition. In particular, the following methods have been chosen: Wiskott's Elastic Bunch Graph Matching (EBGM) [1], eigenfaces (PCA) [24], and Ahonen's Local Binary Patterns (LBP) [25]. The results are shown in Table IV.

TABLE IV. COMPARISON BETWEEN METHODS

Algorithm	Accuracy (%)
EBGM	80.9
PCA	66.4
LBP	74.5
Our proposal	89.7

From these results, we can deduce that connectionist proposals provides a promising method to compete with other well-known methods in face recognition applications.

VI. CONCLUSIONS & FUTURE WORK

This paper has carried out a study on the state of current knowledge on the problem of face recognition focused on the use of the EBGM method. Applying connectionist techniques to build the database of knowledge that make up the collection of faces to recognition has improved the results.

Taking into account that the system is applied in a controlled environment and with a small number of individuals, both the size of the information required and the time spent searching to identify an individual from an image is optimized.

We can highlight here a key feature: Applying a SOM obtains a database more compact than traditional methods. In EBGM based algorithms, the bunch graph on which perform the matching process contains information of each of the graphs obtained for each input image of the training phase, therefore, for more images of individuals larger size of the database. In the proposed method, the information is compacted. It is the map size which determines the number of graphs to be used in the matching phase. Thus, training can be applied to an extensive battery of images without affecting the amount of memory required to store the entire database in memory during execution of the matching algorithm.

Base on the current outcomes, our future work will be unfolded along two directions: one is to make more exhaustive experiments with a great number of images and individuals. The other direction is to prove with other recognition methods in combining with neural network techniques to explore the potential of this approach.

Finally, as demonstrated in the experimental section, the classification efficiencies above 89% can be achieved, leading to optimism on the implementation of the proposed work in a real environment.

REFERENCES

[1] L. Wiskott, J-M. Fellous, N. Krüger, and C. von der Malsburg. “Face Recognition by Elastic Bunch Graph Matching”. IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 19, n. 7, pp. 775-789. Jul. 1997.

[2] C. Kotropoulos, and I. Pitas “Rule-based face detection in frontal views”, IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4, pp. 2537-2540, 1997.

[3] T. Kohonen, “Self-Organising Maps”, 2nd ed., Springer-Verlag, Berlin, 1997.

[4] J. Azorín-López, M. Saval-Calvo, A. Fuster-Guilló, A. Oliver-Albert: A Predictive Model for Recognizing Human Behaviour based on Trajectory Representation. International Joint Conference on Neural Networks, 2014.

[5] A. Rattani, N. Agarwal, H. Mehrotra, and P. Gupta, “An efficient fusion-base.d classifier”. In Workshop on Computer Vision, Graphics and Image Processing (WCVGIP), pp. 104-109, 2006.

[6] L. L. Shen, and L. Bai, “A review on Gabor wavelets for face recognition”, Pattern Anal. Appl., vol. 9, pp. 273 -292, 2006.

[7] D. Monzo, A. Albiol, and J. M. Mossi, “A comparative study of facial landmark localization methods for face recognition using HOG descriptors”. In Proceedings of the 20th International Conference on Pattern Recognition (ICPR), pp. 1330-1333, 2010.

[8] G. Guo, G. Mu, and K. Ricanek, “Cross-Age Face Recognition on a Very Large Database: The Performance versus Age Intervals and Improvement Using Soft Biometric Traits”, 20th International Conference on Pattern Recognition, ICPR, pp. 3392-3395, 2010.

[9] X. Chen, C. Zhang, F. Dong, and Z. Zhou, “Parallelization of elastic bunch graph matching (EBGM) algorithm for fast face recognition”. In Proceedings of the 2013 IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP) pp. 201-205.

[10] A. Khatun, and M. A. A. Bhuiyan, “Neural network based face recognition with Gabor filters”. International Journal of Computer Science and Network Security, vol. 11, pp. 71-74, 2011.

[11] S. Mitra, S. Parua, A. Das, and D. Mazumdar, “A Novel Datamining Approach for performance improvement of EBGM based face recognition system to handle large database”, In Advances in Computer Science and Information Technology, pp. 532-541, 2011.

[12] S. Sarkar, “Skin segmentation based elastic bunch graph matching for efficient multiple face recognition”. In Advances in Computer Science, Engineering & Applications, pp. 31-40, 2012.

[13] Y. T. Li, and J. P. Wachs, “Hierarchical elastic graph matching for hand gesture recognition”. In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, pp. 308-315, 2012.

[14] R. Espí, F. A Pujol, H. Mora, J. Mora, Development of a Distributed Facial Recognition System Based on Graph-Matching, International Symposium on Distributed Computing and Artificial Intelligence, pp. 498, 502, 2008.

[15] D. González-Jiménez, and J. L. Alba-Castro, "Shape-Driven Gabor Jets for Face Description and Authentication," IEEE Trans. Information Forensics and Security, vol. 2, n. 4, pp. 769-780, 2007.

[16] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, “Localizing Parts of Faces Using a Consensus of Exemplars”, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 545-552, 2011.

[17] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, “Real-time facial feature detection using conditional regression forests”. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2578-2585.

[18] T. Baltrusaitis, P. Robinson, and L. Morency, “3D constrained local model for rigid and non-rigid facial tracking”. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2610-2617.

[19] S. Jahanbin, H. Choi, A. C. Bovik, “Passive Multimodal 2-D+3-D Face Recognition Using Gabor Features and Landmark Distances,” Transactions on Information Forensics and Security,IEEE, vol. 6, n. 4, pp. 1287-1304, December 2011.

[20] X. Jin, X. Tan and L. Zhou. “Face Alignment Using Local Hough Voting”. In Proceedings of the 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pp. 1-8, 2013.

[21] H. Yin, “The self-organizing maps: Background, theories, extensions and applications”. In Computational intelligence: A compendium, pp. 715-762, 2008.

[22] J. A. F. Costa, “Clustering and visualizing SOM results”. In Intelligent Data Engineering and Automated Learning–IDEAL 2010, pp. 334-343.

[23] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi, "The FERET evaluation methodology for face recognition algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22 (10), 2000.

[24] M. Turk, and A. Pentland, “Eigenfaces for recognition”. J Cognitive Neurosci, vol. 3, pp. 71-86, 1991.

[25] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face recognition with local binary patterns”. In Pajdla T, Matas J (eds.) Proceedings of the 8th European Conference on Computer Vision, ECCV 2004, Part I, Springer Berlin/Heidelberg, pp. 469-481.

[26] J. Canny, “A Computational Approach to Edge Detection”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, pp. 679-698, 1986.