Automatic face recognition using SIFT and networks of tagged neural cliques

Ehsan Sedgh Gooya, Dominique Pastor and Vincent Gripon

Institut Mines Telecom; Telecom Bretagne; UMR CNRS 6285 Lab-STICC
Email: name.surname@telecom-bretagne.eu

***Abstract*—Bearing information by a fully interconnected subgraphs is recently improved in the neural network of cliques. In this paper, a face recognition system is presented using such networks where local descriptors are used to perform feature extraction. In the wide range of possible image descriptors for face recognition, we focus specifically on the Scale Invariant Feature Transform (SIFT). In contrast to standard methods, our proposed method requires no empirically chosen threshold. Moreover, it performs matching between sets of features, in addition to individual feature matching. Thus, we favor joint occurrences of descriptors during the recognition process. We compare our approach to state of the art face recognition systems based on SIFT descriptors. The evaluation is carried out on the Olivetti and Oracle Research Laboratory (ORL) face database, whose diversity is significant for assessing face recognition methods.**

Keywords–Face recognition, neural networks, associative memories, neural cliques, SIFT descriptors.

I. INTRODUCTION

Instantly recognizing a familiar face is easy task for humans. However, as many processes related to vision, automatic pattern recognition is generally difficult. Face recognition is among the most visible and challenging research topics in computer vision and automatic pattern recognition [1], and many methods, such as Eigenfaces [2], Fisherfaces [3] and SVM [4], have been proposed in the past two decades. Recently the sparse representation (or coding) based classication (SRC) has been successfully used in face recognition [5], [6]. In SRC, the testing image is represented as a sparse linear combination of the training samples, and the representation delity is measured by the $l_1 - norm$ of coding residual.

However, the last word in pattern matching is the human brain in the sense that it seeks to identify links between what it currently observes and what it has experienced in the past.

Over the last ten years, much attention has been given to feature-based methods such as SIFT [7]. This is due to the fact these descriptors remain invariant under rotation, scaling and variation in lightning condition. In a conventional method, SIFT features are extracted from all the faces in the database. Then, given a query face image, each feature extracted from that face is compared to those of each face in the database. A query feature is considered to match one of the database according to a certain threshold-based criterion. The face in the database with the largest number of matched descriptors is considered as the nearest face.

Although the nearest face criterion may give very good results, it suffers from the following limitations. To begin with, only the first nearest neighbors are used to characterize the contents of the database. Also, the threshold set by the user is obtained a posteriori and, as such, varies from one experiment to another.

To overcome these drawbacks, we propose a novel approach based on matching sets of descriptors. This approach relies on a new extension of the neural network introduced in [8] and [9] that embeds messages to learn into cliques. Basically, this neural network is an associative memory (denoiser). However, to the best of our knowledge, it is the first time that it is used for pattern recognition.

The reason why we investigate using clique networks with SIFT descriptors is because of their error correcting capability. Intuitively, the mismatches that may occur when pairing SIFT descriptors may be corrected by the redundancy of clique patterns in the neural network.

The face recognition method proposed in this paper combines SIFT features and networks of neural cliques. In the course of the paper, the fundamental concepts of the SIFT algorithm are presented in Section II. The third section (Section III) reviews different sift matching methods for face recognition. The neural network of neural cliques is described in Section IV and the whole face recognition system based on neural network of neural cliques is presented in Section V. Finally, we present and discuss the results of the proposed face recognition system in Section VI.

II. SCALE INVARIANT FEATURE TRANSFORM (SIFT)

The Scale Invariant Feature Transform algorithm was proposed by David G. Lowe in [7] and extracts distinctive features. These features are invariant to rotation, scaling and partly invariant to changes in illumination and affine transformation of images. Therefore, these features are good candidates for face recognition. The main steps to calculate the SIFT features of an image are the following ones.

A. Keypoint localization

To efficiently detect stable keypoint locations, scale-space extrema in the difference-of-Gaussian (DoG) are used during the computation of the SIFT descriptors. The scale-space is defined as a function $L(x, y, \sigma)$ obtained by Gaussian kernel convolution with the input image so that:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{1}$$

where $I(x, y)$ is the input image and $G(x, y, \sigma)$ is the Gaussian function:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{x^2+y^2}{2\sigma^2}} \ . \tag{2}$$

 ISBN: 978-1-61208-390-2

$DoG\,(x, y, \sigma)$ can be computed from the difference of two nearby scales separated by an empirically chosen constant multiplicative factor k:

$$DoG(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ = L(x, y, k\sigma) - L(x, y, \sigma)\,. \quad (3)$$

The efficient approach to construction of $DoG(x, y, \sigma)$ is shown in Figure 1.

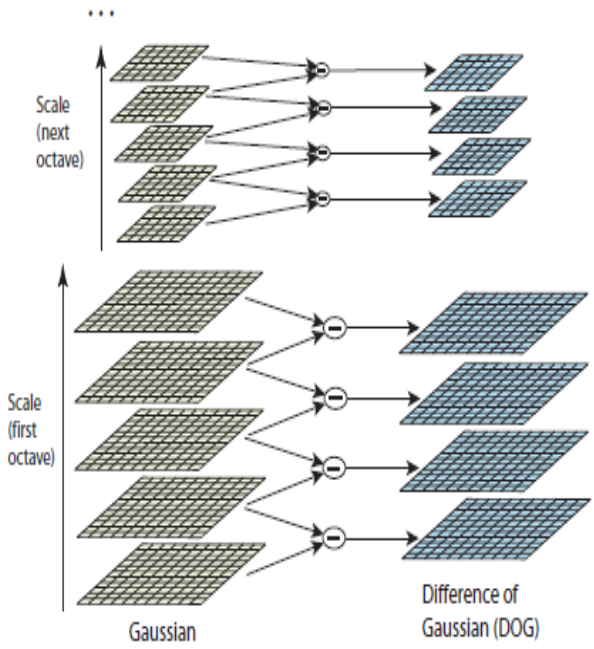

Figure 1. Excerpt from [7]). The convolved images are grouped by octave (an octave corresponds to doubling the value of σ), and the value of k_i is selected so that we obtain a fixed number of convolved images per octave. Then the Difference-of-Gaussian (DoG) images are taken from adjacent Gaussian-blurred images per octave.

In order to detect the local maxima and minima of $DoG(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and to its nine neighbors in the scales above and below, as shown in Figure 2. After comparison, the sample is selected only if it is larger than all of these neighbors or smaller than all of them. Moreover, the algorithm eliminates candidates that are located on an edge or have poor contrast.

B. Assigning Rotation to Keypoint

Given a keypoint at position (x_0, y_0) for a given scale σ_0, the gradient principal direction must be computed. To do so, for each pixel (x, y) directly connected to x_0, y_0 we compute the magnitude $m(x, y)$ and orientation $\theta(x, y)$ as follows:

$$m(x, y) = \\ \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \\ \theta(x, y) = \tan^{-1} \frac{L(x, y-1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\,. \quad (4)$$

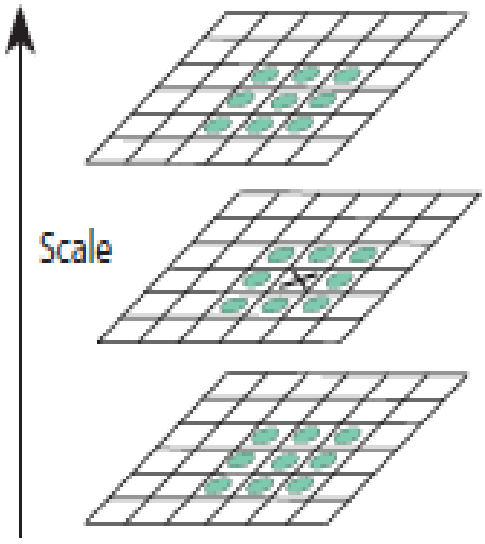

Figure 2. (Excerpt from [7]). Maxima and minima of the DoG images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3×3 regions at the current and adjacent scales (marked with circles).

An orientation histogram with 36 bins covering 360 degrees is formed from the gradient orientations of the sample points around the keypoint. Each sample added to the histogram is weighted by its gradient magnitude. Then, the maximum orientation is assigned to this keypoint. For any other orientation within 80% of the maximum orientation, a new keypoint is created with this orientation. Each keypoint is rotated in direction of its orientation and then normalized. The maximum orientation, θ_0, is assigned to the keypoint. A keypoint is then entirely determined given the four parameters $(x_0, y_0, \sigma_0, \theta_0)$.

C. Construction of the feature descriptors

The 4×4 subregions located around a given keypoint are delimited, each containing 4×4 pixels. In each subregion, the orientations and magnitudes at each pixel are calculated. An orientation histogram of 8 bins is computed for each subregion. The corresponding gradient values are weighted by a Gaussian circular window. The 16 resulting histograms are then normalized and form a vector with 128 dimensions (16×8).

Figure 3 shows an example of a keypoint with its descriptor and orientation.

III. Review of SIFT-based matching methods for face recognition

A. Aly's matching

In [10], each SIFT descriptor in the test image is compared with every descriptor of each training image. The comparison is performed using cosine similarity of two feature $\mathbf{f_1}$ and $\mathbf{f_2}$ computed as follows:

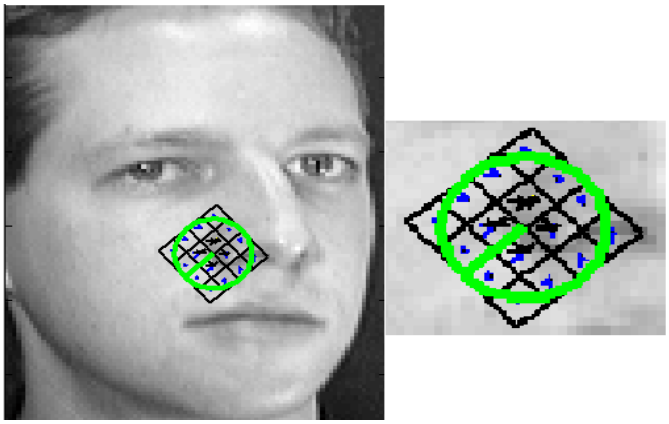

Figure 3. Example of a SIFT descriptor and orientation.

$$\Omega(\mathbf{f_1}, \mathbf{f_2}) = \frac{\mathbf{f_1} \times \mathbf{f_2}}{\|\mathbf{f_1}\| \times \|\mathbf{f_2}\|} \ . \tag{5}$$

A feature $\mathbf{f}$ from the query face image is considered to match a feature $\mathbf{f_1}$ from the gallery images if a) $\mathbf{f_1}$ is the most similar feature to $\mathbf{f}$ in the gallery images and b) the second closest feature $\mathbf{f_2}$ to $\mathbf{f}$ in the gallery images is such that $\Omega(\mathbf{f_2}, \mathbf{f}) - \Omega(\mathbf{f_1}, \mathbf{f}) \geq \Omega_{\min}$, where $\Omega_{\min}$ is a fixed threshold.

B. Lenc-Kral and Kepenekci's matching

In [11], for each feature of the query face image, the most similar feature of the gallery face is identified. The sum of the highest similarities is computed and is used as a measure of similarity between two faces. Kepenekci's SIFT matching combines two methods of matching and uses a weighted sum of the two values as a result. The cosine similarity is employed for feature comparison.

C. Support Vector Machine classifier

The face recognition method presented in [12] employs SIFT features to extract discriminative local features and Support Vector Machine (SVM) as a classifier. Basically, SVM is able to separate positive and negative examples using decision surfaces constructed by optimal separating hyperplanes.

IV. Networks of Tagged Neural Cliques

An associative memory is a device capable of storing vectors, then retrieving them when some coordinates are missing or altered. Recently, an implementation of an associative memory based on neural networks was proposed in [8]. This network can store a large number of binary vector patterns and retrieve them with low error probability and high memory efficiency, even in case of erasures. The principle of this model is to embody vectors into fully interconnected subgraphs called cliques. Contrary to the celebrated Hopfield model [13], connections are binary in [8].

However, the vectors that can be handled by such a network are too constrained for our application in face recognition. For this reason, we follow the extension proposed in [9], in which any binary vector can be handled by the network. We extend the functionalities of this model so as to perform classification of vector patterns for face recognition.

As any associative memory, two operations are performed by the network: storing and retrieving. In the following subsections, we describe these two operations.

A. Storage (learning) process

In this paper, the binary neural network contains n neurons. This network stores c gallery patterns. Each gallery pattern $\mathbf{g}_k$ is defined as the concatenation between pattern vector $\mathbf{x}_k$ with dimension d and its associated class represented by vector $\mathbf{e}_k$ (k-th element of the canonical base in $\mathbb{R}^{\mathrm{c}}$):

$$\mathbf{g}_k = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{e}_k \end{pmatrix} \in \{0;1\}^{d+c} \ . \tag{6}$$

Let us index these neurons from 1 to $n = d + \mathrm{c}$. The underlying graph is fully represented by its adjacency binary matrix $\boldsymbol{W}$ of size $n \times n$, in which the gallery patterns are stored.

In the storage procedure, the first d neurons are employed to embody the vectors $\mathbf{x}_k$ in the form of cliques and the c remaining neurons are used to tag the cliques (see figure 4). Accordingly, $\mathbf{e}_k$ is the binary unit vector whose unique coordinate equal to 1 is the tag index associated with $\mathbf{x}_k$. Denote by $(\mathbf{g}_k)_{1 \leq k \leq \mathrm{c}}$ the sequence of gallery vectors to be stored. Then $\boldsymbol{W}$ is defined as:

$$\boldsymbol{W} = \max_k \left(\mathbf{g}_k \cdot \mathbf{g}_k^{\mathrm{T}} \right) \tag{7}$$

where $(\bullet)^{\mathrm{T}}$ is the transpose operator. Using this process, the connection between neurons i and j is set to 1 if there exists k such that $\mathbf{g}_k(i) = \mathbf{g}_k(j) = 1$.

According to the foregoing, the computation of the adjacency matrix $\boldsymbol{W}$ is independent of the order in which gallery vectors are presented. Moreover, adding a new gallery pattern can be done online, independently of previously stored patterns.

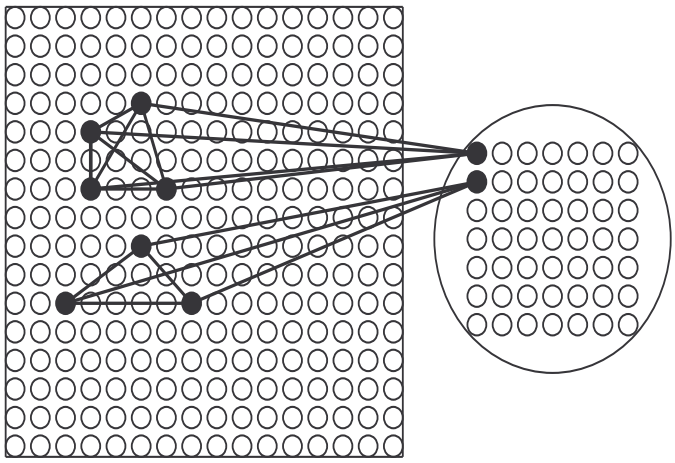

Figure 4. Example of two tagged cliques: the first d neurons on the left are used to embody the vectors $\mathbf{x}_k$ in the form of cliques and the c neurons on the right are employed to tag the cliques.

 ISBN: 978-1-61208-390-2

B. Retrieving process

The retrieving algorithm is a two-step and possibly iterative procedure. The first step aims at matching the input with a similar clique in the neural network. The purpose of the second step is to retrieve the associated label. To perform retrieval, we use a nonlinear filter f that operates over a vector $\mathbf{v}$. It consists in putting to zero all the coordinates that are not maximum and to one those that reach the maximum:

$$f(\mathbf{v})_i = \begin{cases} 1 & \text{if } \mathbf{v}_i = \max_j \mathbf{v}_j \\ 0 & \text{otherwise} \end{cases} . \quad (8)$$

To ease the reading, we also introduce the operator $\pi_{\mathbf{x}}(\mathbf{v})$ (resp. $\pi_{\mathbf{e}}(\mathbf{v})$) that extracts the vector made of the first (resp. last) d (resp. c) coordinates of $\mathbf{v}$. Conversely, we denote by $(\mathbf{v};\mathbf{v}')$ the concatenation of vectors $\mathbf{v}$ and $\mathbf{v}'$. Finally, we denote by $\mathbf{0}^{\mathrm{c}}$ the zero vector with dimension c. Algorithm 1 is used to classify $\mathbf{x}$.

Algorithm 1: Classification algorithm with neural network of tagged cliques.

Input: Input pattern $\mathbf{x}$ and adjacency matrix $\boldsymbol{W}$
Output: $\widehat{\mathbf{e}}_k$, the class indicator vector estimated for $\mathbf{x}$

1 $\widehat{\mathbf{x}} = \pi_{\mathbf{x}}\left(f(\boldsymbol{W}\begin{pmatrix}\mathbf{x}\\ \mathbf{0}^{\mathrm{c}}\end{pmatrix})\right)$;

2 $\widehat{\mathbf{e}}_k = \pi_{\mathbf{e}}\left(f(\boldsymbol{W}\begin{pmatrix}\widehat{\mathbf{x}}\\ \mathbf{0}^{\mathrm{c}}\end{pmatrix})\right)$;

If the output is not a unit vector then we consider that the classification failed. Otherwise, the nonzero coordinate is our estimator for the class associated with $\mathbf{x}$.

V. Face recognition using binary networks of tagged neural cliques

A. Storing (Learning) face images in binary networks of neural tagged cliques

Let us consider a set of training face images $\mathrm{S} = \{\mathrm{S}_i\}_{i=1}^{L}$ of cardinality L. We denote by $c \leq L$ the number of distinct persons (classes). For each person k, we compute the set F_k of SIFT features of all their corresponding images. We then index the set of all features $\mathrm{F} \triangleq \bigcup_{1\leq k\leq \mathrm{c}} \mathrm{F}_k = \{\mathbf{f}_1, \ldots, \mathbf{f}_d\}$.

We define the gallery vectors $\mathbf{g}_k$ by choosing $\mathbf{x}_k$ as the indicator vector of the subset F_k:

$$(\mathbf{x}_k)_i = \begin{cases} 1 & \text{if } \mathbf{f}_i \in \mathrm{F}_k \\ 0 & \text{otherwise} \end{cases} . \quad (9)$$

We afterwards perform the storage according to the method described in subsection IV-A. An example of a clique is shown in Figure 6. Such a graphical pattern is redundant and offer error correcting capabilities [8].

B. Retrieving face images in binary networks of neural tagged cliques

Let $\bar{\mathrm{S}} = \{\bar{\mathrm{S}}_i\}_{i=1}^{\bar{L}}$ be a set of face images to test. These images are novel but correspond to persons already seen in the gallery. First, each test face image $\bar{S}_i$ is described as a set of SIFT features $\bar{\mathrm{F}}_i$ We use the cosine similarity to compare two SIFT features as in (5). The input indicator vector $\mathbf{x}$ of the subset F is then defined as follows:

$$(\mathbf{x})_i = \begin{cases} 1 & \text{if } \exists\, \mathbf{f} \in \bar{\mathrm{F}}_i \text{ such that } i = \underset{j}{\operatorname{argmin}}\ \theta(\mathbf{f}, \mathbf{f}_j) \\ 0 & \text{otherwise} \end{cases} . \quad (10)$$

for i from 1 to d. According to the two-step Algorithm 1, the input pattern $\mathbf{x}$ is then retrieved and classified.

VI. Experimental evaluation

The Olivetti and Oracle Research Laboratory (ORL) face database is used in order to test our method in the presence of headpose variations. There are 10 different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying lighting, facial expressions (open / closed eyes, smiling / not smiling), facial details (glasses / no glasses) and head pose (tilting and rotation up to 20 degrees). All the images were taken against a dark homogeneous background. Figure 5 shows the whole set of 40 individuals, 1 images per person from the ORL database.

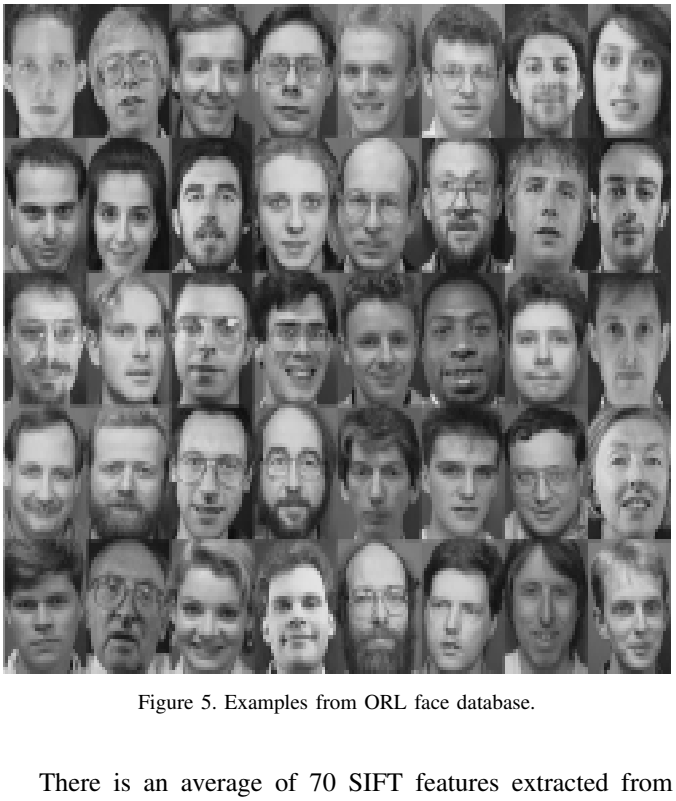

Figure 5. Examples from ORL face database.

There is an average of 70 SIFT features extracted from each image using the implementation proposed in [14]. Twenty independent runs were carried out. In each run, the dataset is randomly split into two halves, one for training (K images per class) and one for testing (the remaining $10 - K$ images per class).

Table I displays the results (average on the runs) obtained on the ORL database by several state of the art approaches, for comparison to the method proposed in this paper. All these methods are based on SIFT features. SIFT-based face recognition methods are actually more robust than other ones [10]. As shown in this table, the face recognition method introduced in this paper outperforms previously proposed approaches.

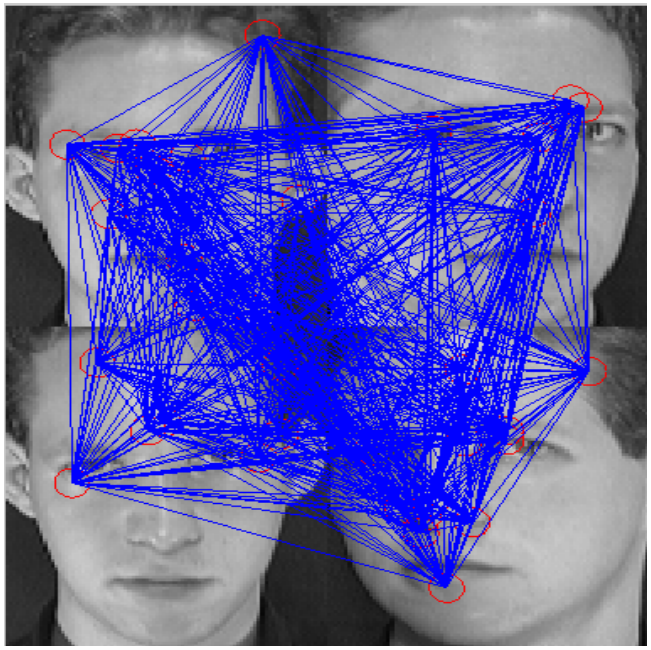

Figure 6. On this images, points of interests (represented as neurons in our model) have been fully interconnected to obtain a clique.

TABLE I. RECOGNITION RATES OF DIFFERENT MATCHING SCHEMES FOR THE ORL DATASET WITH RESPECT TO DIFFERENT SIZES FOR THE TRAINING SET.

Method	**Number of training images**			
	K = 5	K = 6	K = 7	K = 8
SIFT-SVM [12]	N/A	N/A	95.6	97.4
Aly [10]	92.42	95.27	96.88	98.36
Lenc-Kral [11]	96.75	97.86	98.65	98.86
Kepenekci [11]	97.92	97.86	98.65	99.17
Proposed method	**98.82**	**99.55**	**99.71**	**99.88**

It is worth pointing out that the decoding of a pattern in the clique network can be efficiently parallelized [15], [16].

VII. CONCLUSION AND FUTURE WORK

We successfully transformed the associative memory introduced in [9] into a classifier. Thanks to the error correcting properties of such memory, our proposed method outperforms state of the art SIFT-based face recognition approaches on the ORL database, without having to blow up the number of neurons. Since all the face recognition methods are based on the same feature descriptors, our results emphasize the interest of using clique-based networks as classifiers. It is worth pointing out that our method requires no threshold for face recognition, in contrast to the other ones.

Regarding scalability, future work involves assessing the impact of reducing network size towards performance. Compared to the theory of grandmother cells where a piece of information is carried out by a unique neuron, cliques offer the possibility to encode such data as an assembly of units. As a consequence the number of units needed to store information is significantly reduced in clique networks compared to perceptrons [17], for instance. Therefore one may expect complexity reduction compared to exhaustive search.

Regarding performance, we consider using complementary features to improve robustness of the recognition. In this respect, combining local SIFT descriptors with global features should increase accuracy of the system.

ACKNOWLEDGMENT

This work was partially funded as part of the NEUCOD project by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement.

REFERENCES

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” Acm Computing Surveys (CSUR), vol. 35, no. 4, 2003, pp. 399–458.

[2] M. Turk and A. Pentland, “Eigenfaces for recognition,” Journal of cognitive neuroscience, vol. 3, no. 1, 1991, pp. 71–86.

[3] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, 1997, pp. 711–720.

[4] B. Heisele, P. Ho, and T. Poggio, “Face recognition with support vector machines: Global versus component-based approach,” vol. 2. IEEE, 2001, pp. 688–694.

[5] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, 2009, pp. 210–227.

[6] L. Li, S. Li, and Y. Fu, “Learning low-rank and discriminative dictionary for image classification,” Image and Vision Computing, vol. 32, no. 10, 2014, pp. 814–823.

[7] D. Lowe, Object Recognition from Local Scale-Invariant Features. Springer Boston, 1999.

[8] V. Gripon and C. Berrou, “Sparse neural networks with large learning diversity,” IEEE Transactions on Neural Networks, vol. 22, no. 7, July 2011, pp. 1087–1096.

[9] B. K. Aliabadi, C. Berrou, V. Gripon, and X. Jiang, “Learning sparse messages in networks of neural cliques,” IEEE Transactions on Neural Networks and Learning Systems, August 2012.

[10] M. Aly, “Face recognition using sift features,” CNS/Bi/EE report, vol. 186, 2006.

[11] L. Lenc and P. Král, “Novel matching methods for automatic face recognition using sift,” Artificial Intelligence Applications and Innovations, 2012, pp. 254–263.

[12] L. Zhang, J. Chen, Y. Lu, and P. Wang, “Face recognition using scale invariant feature transform and support vector machine,” ICYCS’08: The 9th International Conference for Young Computer Scientists, 2008, pp. 1766–1770.

[13] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” Proceedings of the national academy of sciences, vol. 79, no. 8, 1982, pp. 2554–2558.

[14] A. Vedaldi, “An open implementation of the SIFT detector and descriptor,” no. 070012, 2007.

[15] H. Jarollahi, N. Onizawa, V. Gripon, and W. J. Gross, “Architecture and implementation of an associative memory using sparse clustered networks,” in IEEE International Symposium on Circuits and Systems (ISCAS), 2012, pp. 2901–2904.

[16] B. Larras, C. Lahuec, M. Arzel, and F. Seguin, “Analog implementation of encoded neural networks,” in IEEE International Symposium on Circuits and Systems (ISCAS), 2013, pp. 1612–1615.

[17] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.” Psychological review, vol. 65, no. 6, 1958, pp. 386–408.

 ISBN: 978-1-61208-390-2