

# Local Representations and a direct Voting Scheme for Face Recognition

R. Paredes, J. C. Pérez, A. Juan, and E. Vidal

Universitat Politècnica de València,  
Institut Tecnològic d'Informàtica,  
Departament de Sistemes Informàtics i Computació,  
Camí de Vera, s/n, 46022 València, Spain.  
`rparedes@iti.upv.es`

**Abstract.** A new approach combining a *simple local representation method* with a *k*-nearest neighbours-based *direct voting scheme* is proposed for face recognition. This approach rises computational problems that we effectively solve through an approximate fast *k*-nearest neighbours search technique. Experimental results with the widely used Olivetti Research Ltd (ORL) face database are reported showing the effectiveness of the proposed approach.

*Key words:* Face Recognition, Local Features, Nearest Neighbour

## 1 Introduction

Among biometric identification systems, *Face recognition* is one of the most unobtrusive methods, well tolerated by users, and with a wide range of applications. In a typical recognition scenario, the system has to show a high degree of robustness to illumination, viewpoint, background, gesture and facial details. Therefore, a powerful classification method and a specially devised feature extraction technique are essential to achieve good performance in this task. We propose a procedure based on *local representations*, which has shown promising preliminary results.

Local representations are often used combined with other schemes. They are explicitly cited mainly in the image database retrieval literature [1–3], where invariances to scale, rotation and illumination changes are needed (translation invariance is generally not required, since each local feature is located at a particular point in the image). In most face recognition tasks, rotation invariance is not needed, and scale and illumination normalization can be applied in the preprocessing stage.

In a classical classifier, each object is represented by a feature vector, and a discrimination rule is applied to classify a test vector that also represents one object. Local representation, however, implies that each image is scanned to compute many feature vectors. Each of these vectors could be classified into a

different class, and therefore a decision scheme is required to finally decide a single class for a test image.

In this paper, a combination of a simple local representation method with a direct decision scheme based on  $k$ -nearest neighbours is proposed. This approach has shown to be most effective if the set of local feature vectors from the training data is large. The high computational cost is avoided by using an approximate fast  $k$ -nearest neighbours search technique.

## 2 Proposed approach

### 2.1 Preprocessing and feature extraction

Preprocessing consists of two steps. The first step aims at *selecting* pixels with high information content. Although a number of methods exist to detect such pixels [4], most of them are not appropriate for face images since textured areas are not frequent in this type of images. Therefore, we have chosen a simple and fast method: the local variance in a small window is measured for each pixel and those pixels having local variance above a certain global threshold are selected. The second preprocessing step consists in performing a simple gradient operation to reduce the variability in illumination conditions. More specifically, a *gradient image* is obtained by computing the average of the horizontal and vertical gradients at *each selected pixel*. A preprocessing example is shown in fig. 1.

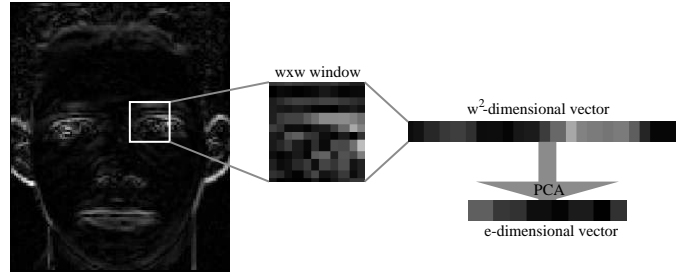


**Fig. 1.** Preprocessing example. From left to right: original image, local variance image, thresholded variance image with selected pixels in white, and gradient image.

As discussed in section 1, our feature extraction technique represents each image by many feature vectors belonging to different regions of the image. These regions correspond to the pixels selected by local-variance thresholding. For each of these pixels, a  $w^2$ -dimensional vector of grey values is first obtained in the gradient image by application of a  $w \times w$  window around it. The dimension of the resulting vectors is then reduced from  $w^2$  to  $e$  using *Principal Component Analysis* (PCA), thus obtaining a compact local representation of a region of the image. We label each vector with an identifier of the class, i.e. the person whose face is represented in the image.

Feature extraction can thus be formally stated as follows: Let  $I = \{I_1, \dots, I_n\}$  be a training set of  $n$  face images belonging to  $d$  different persons. For each image  $I_i$ ,  $m_i$   $w^2$ -dimensional feature vectors are obtained. Then, an  $e$ -dimensional eigenbase is computed from the covariance matrix of *all* feature vectors obtained from  $I$ , and these vectors are projected into the space spanned by this eigenbase. Let  $X_i = \{\mathbf{x}_{i1}, \dots, \mathbf{x}_{i,m_i}\}$  be the set of  $e$ -dimensional vectors associated with image  $I_i$ , and let  $T = \cup_{i=1}^n X_i$  be the global set of vectors. Each vector  $\mathbf{x}_i$  has an associated class label  $\omega^i \in \{\omega_1, \dots, \omega_d\}$  which is the class label of the image  $I_i$ . Obviously, all the images belonging to the same person share the same class label.

Feature extraction for test images is done in the same way, by using the eigenbase computed from the training set. This is illustrated in fig. 2.



**Fig. 2.** Feature extraction process. A  $w \times w$  window is obtained for each pixel in the gradient image exceeding the variance threshold in the preprocessing phase (white pixels in third picture of figure 1). The  $w^2$  vector associated with that window is mapped into  $R^e$ ,  $e \ll w^2$ , using Principal Component Analysis.

## 2.2 Classification through a $k$ -NN based Voting Scheme

The Classification procedure used in this work is closely related to a family of techniques often referred to as “*direct voting schemes*” [2]. It is in fact based on applying the well known  $k$ -Nearest Neighbour rule to the set of vectors representing a test image, using the global vector set  $T$  as the reference or prototype set. More formally, we can present the proposed classification technique under the statistical framework of “*Classifier Combination*” [5].

Let  $Y$  be a test image. Following the conventional probabilistic framework,  $Y$  can be optimally classified in a class  $\hat{\omega}$  having the maximum posterior probability:

$$\hat{\omega} = \arg \max_{1 \leq j \leq d} P(\omega_j | Y) \quad (1)$$

By applying the feature extraction process described in the previous section to  $Y$ , a set of  $m_Y$  feature vectors,  $\{\mathbf{y}_1, \dots, \mathbf{y}_{m_Y}\}$  is obtained. Thus, we can see the classifier (1) as a *combination of  $m_Y$  classifiers*, each for every feature vector of  $Y$ . To this end, assuming independence between each  $\mathbf{y}_i$ ,  $P(\omega_j|Y)$  can be written as the product of the posterior probabilities associated to every feature vector and (1) becomes:

$$\hat{w} = \arg \max_{1 \leq j \leq d} \prod_{i=1}^{m_Y} P(\omega_j|\mathbf{y}_i) = \arg \max_{1 \leq j \leq d} \sum_{i=1}^{m_Y} \log P(\omega_j|\mathbf{y}_i)$$

This is commonly called the “*product rule*” for classifier combination [5]. An important drawback of this rule is that small probabilities (which often are poorly estimated) tend to dominate the combination result, yielding poor overall estimates of  $P(\omega_j|Y)$  and generally leading to low classification performance in practice. In order to alleviate these problems the so called “*sum rule*” is often preferred in practical applications. This rule can be seen as a way of smoothing the effect of (poorly estimated) small probabilities.

In real situations, most posterior probabilities  $P(\omega_j|\mathbf{y}_i)$  generally have values either close to 1 or close to 0. For those which are close to 1, a good linear approximation of their logarithms is:

$$\log P(\omega_j|\mathbf{y}_i) \approx P(\omega_j|\mathbf{y}_i) - 1,$$

Obviously, for those probabilities which are close to 0, this linear approximation is not so good, yielding values significantly larger than the correct ones. Nevertheless, the error so introduced actually has a beneficial *smoothing* effect which tends to automatically compensate the generally poor estimates of such small probabilities. All in all, by using this linear approximation, (1) can be now written as:

$$\hat{w} = \arg \max_{1 \leq j \leq d} \sum_{i=1}^{m_Y} P(\omega_j|\mathbf{y}_i) \quad (2)$$

which corresponds to the above mentioned “*sum rule*” for classifier combination.

In our case, posterior probabilities are directly estimated by  $k$ -Nearest Neighbours. Let  $k_{ij}$  the number of neighbours of  $\mathbf{y}_i$  belonging to class  $\omega_j$ . Assuming the average number of reference vectors representing all the training images of each class is more or less constant, a well known estimate of  $P(\omega_j|\mathbf{y}_i)$  is:

$$\hat{P}(\omega_j|\mathbf{y}_i) = \frac{k_{ij}}{k}$$

and, using this estimate in (2), our classification rule becomes:

$$\hat{w} = \arg \max_{1 \leq j \leq d} \sum_{i=1}^{m_Y} k_{ij} \quad (3)$$

That is, a class  $\hat{w}$  with the largest number of “*votes*” accumulated over all the vectors belonging to the test image is selected. This justifies why techniques of this type are often referred to as “*voting schemes*”.

### 2.3 Efficient approximate search of matching feature vectors

An important feature of a practical face recognition method is speed. The pre-processing and feature extraction steps described in section 2.1 have been carefully chosen to be simple and fast, but the retrieval of a large number of high-dimensional vectors for each test image, from a huge pool of vectors obtained from the reference images seems an intractable issue.

A first recourse could be to use a large subsampling value in the reference images, in order to store as few vectors as possible, and also in the test image, to perform a small number of searches. A subsampling factor  $s$  implies examining only one in every  $s$  columns and rows of the image, therefore reducing by a factor of  $s^2$ , on average, the number of points of interest that give rise to feature vectors. However, a large subsampling value in the reference images significantly reduces the chances that a test feature vector is found in the reference data set, and the performance rapidly degrades when  $s$  is increased. Still, a large subsampling value in the test image is acceptable. While this entails a reduction in the number of queries to be performed, there is no loss of accuracy in any single query.

A consequence of the previous discussion is that a large reference set is essential for the proposed method to be effective. Therefore, a fast search algorithm has to be applied to perform the complete process in a reasonable time. To this end, we have adopted the well known *kd-tree* data structure. It is a binary tree where each node represents a region in a  $k$ -dimensional space. Each internal node also contains a hyperplane (a linear subspace of dimension  $k-1$ ) dividing the region into two disjoint sub-regions, each inherited by one of its children. Most *kd-tree* construction procedures divide the regions according to the points that lay in them. This way, the hierarchical partition of the space can either be carried out to the last consequences to obtain, in the leaves, regions with a single point in them, or can be halted in a previous level so as each leaf node holds  $b$  points in its region.

In a *kd-tree*, the search of the nearest neighbour of a test point is performed starting from the root, which represents the whole space, and choosing at each node the sub-tree that represents the region of the space containing the test point. When a leaf is reached, an exhaustive search of the  $b$  prototypes residing in the associated region is performed. However, the process is not complete at this point. Since it is possible that among the regions defined by the initial partition, the one containing the test point be not the one containing the nearest prototype. It is easy to determine if this can happen in a given configuration, in which case the algorithm backtracks as many times as necessary until it is sure to have checked all the regions that can hold a prototype nearer to the test point than the nearest one in the original region. The resulting procedure can be seen as a Branch-and-Bound algorithm.

If a guaranteed exact solution is not needed, as can be assumed in our case, the backtracking process can be aborted as soon as a certain criterion is met by the current best solution. In [9], the concept of  $(1 + \epsilon)$ -approximate nearest neighbour query is introduced. A point  $p$  is a  $(1 + \epsilon)$ -approximate nearest neigh-

bour of  $q$  if the distance from  $p$  to  $q$  is less than  $1 + \epsilon$  times the distance from  $p$  to its nearest neighbour. This concept has been used here to obtain an efficient approximate search that can easily cope with very large sets of reference vectors.

### 3 Experiments

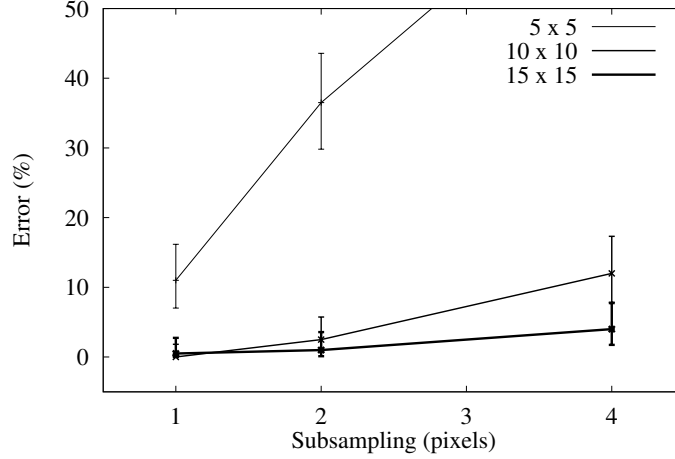
Experiments were carried out with the widely used Olivetti Research Ltd. (ORL) database of faces [10]. This database comprises 400 images of 40 individuals (10 images per individual). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position. In some cases, the images were taken at different times, varying the lighting, facial expressions and details. The size of each image is  $92 \times 112$  pixels, with 256 grey levels per pixel.

Preprocessing and feature extraction of each image was done as described in section 2.1. More concretely, a  $w \times w$  window was centred in each pixel to select those pixels having local variance above a global threshold of 16. Then, each selected pixel in the gradient image was represented as a 40-dimensional feature vector by application of PCA.

Estimation of the classification error rate was done with the same experimental procedure that has been used to estimate the error rate of other approaches tested on the ORL database. The procedure is a simple hold-out with five images of each subject used for training and the other five for testing. This gives a total of 200 training images and 200 test images. Each test image was classified by first computing the 15-nearest neighbours of each of its corresponding feature vectors and then deciding in accordance with the voting scheme described in section 2.2. More precisely, a 3-approximate ( $\epsilon = 2.0$ ) 15-nearest neighbours searching procedure was used instead of an exact technique so as to speed the search up (see section 2.3). This value of  $\epsilon$  offers high search speed at the expense of an insignificant loss of accuracy. Moreover, a subsampling factor  $s$  was used in the reference images, to store as few vectors as possible. Similarly, a subsampling factor of 2 was used in each test image, to reduce the number of searches.

The complete experiment (preprocessing and feature extraction followed by error rate estimation) was carried out for each window size  $w \in \{5, 10, 15\}$  and each subsampling factor  $s \in \{1, 2, 4\}$ . Figure 3 shows each estimated error rate along with its corresponding 95% confidence interval.

From the results in Figure 3, it can be seen that both the window size and the subsampling factor have a great impact in classification performance. As expected, the classification error rate rapidly degrades when the window size is decreased or the subsampling factor is increased. Although the best result corresponds to a window size of 10 and no subsampling (0.0%), a slightly worse yet interesting result is obtained with a window size of 15 and a subsampling factor of 2 (1.0%). Classification of each test image takes about *five* seconds on a Pentium II 450 Mhz processor with the former parameter values. In contrast, only *one* second of CPU time is needed on average when the latter parameter values are used.

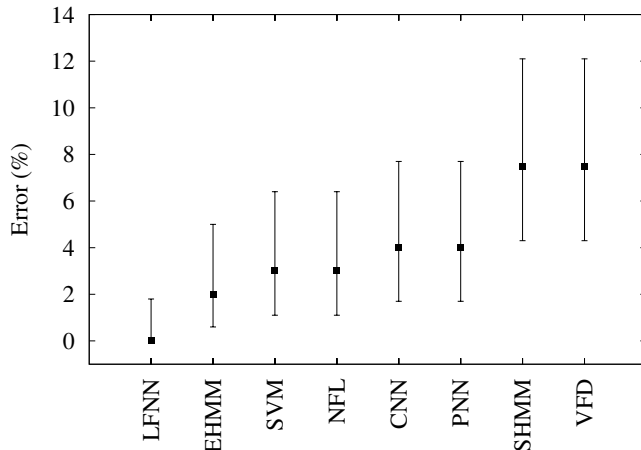


**Fig. 3.** Error rate (%) and its corresponding 95% confidence interval as a function of the subsampling factor in the reference images, for varying window sizes.

Table 1 compares our best error rate and its corresponding 95% confidence interval with those obtained by other successful approaches tested on the ORL dataset. The results in this table are also plotted in fig. 4 for better clarity. Clearly, the proposed approach outperforms those based on Volumetric Frequency Domain Representation [11] and Standard Hidden Markov Models [12]. Convolutional and Probabilistic Neural Networks [13,14] are also improved up to a high level of confidence. Only Nearest Feature Line [15], Support Vector Machines [16] and Embedded Hidden Markov Models [17] give similar results. Our approach, however, is much simpler and still maintains a clear advantage in terms of error rate.

**Table 1.** Error rates and their corresponding 95% confidence intervals for the proposed and other approaches tested on the ORL dataset.

Approach	Error rate	95% conf. interval
Local Features-based Nearest Neighbour	0.0	0.0 — 1.8
Embedded Hidden Markov Models [17]	2.0	0.6 — 5.0
Support Vector Machines [16]	3.0	1.1 — 6.4
Nearest Feature Line [15]	3.0	1.1 — 6.4
Convolutional Neural Networks [13]	4.0	1.7 — 7.7
Probabilistic Neural Networks [14]	4.0	1.7 — 7.7
Standard Hidden Markov Models [12]	7.5	4.3 — 12.1
Volumetric Frequency Domain Rep. [11]	7.5	4.3 — 12.1



**Fig. 4.** Error rates and their corresponding 95% confidence interval for the proposed approach (LFNN=“Local Features-based Nearest Neighbour”) and other classification techniques tested on the ORL dataset (EHMM=“Embedded Hidden Markov Models”, SVM=“Support Vector Machines”, NFL=“Nearest Feature Line”, CNN=“Convolutional Neural Networks”, PNN=“Probabilistic Neural Networks”, SHMM=“Standard Hidden Markov Models” and VFD=“Volumetric Frequency Domain Representation”).

## 4 Conclusions

A novel approach is proposed for face recognition which combines a simple local representation method with a direct voting scheme based on  $k$ -nearest neighbours. This approach rises computational problems that we effectively solve through an approximate fast  $k$ -nearest neighbours search technique. Experimental results with the ORL face database are reported showing the effectiveness of the proposed approach.

Current work is under way to test the proposed approach on other public databases. We are also interested in studying other voting schemes. An attractive possibility is the use of global or semi-local constraints that exclude, for instance, feature vectors located in a very different position in the reference image. These vectors could receive weights depending on the distance, or pondered according to the relations in a graph [6] or other geometric structure. In [1] semi-local constraints are found to improve the results significantly. In the 3D reconstruction literature, matching of stereo images has been also approached through local features and semilocal and global constraints [7, 8]. In [3], the tradeoff between local and global characterization in medical image retrieval is studied.

*Acknowledgement.* Work supported by the Spanish “Ministerio de Ciencia y Tecnología” under grant TIC2000-1703-CO3-01.



## References

1. C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. on PAMI*, 19(5):530–535, 1997.
2. R. Mohr, S. Picard, and C. Schmid. Bayesian decision versus voting for image retrieval. In *Proc. of the CAIP-97*, 1997.
3. C. Shyu et al. Local versus Global Features for Content-Based Image Retrieval. In *Proc. of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 30–34, 1998.
4. R. Deriche and G. Giraudon. A Computational Approach to Corner and Vertex Detection. *Int. Journal of Computer Vision*, 10:101–124, 1993.
5. R.P Duin J. Kittler, M. Hatef and J. Matas. On combining classifiers. *IEEE Trans. on PAMI*, 1998.
6. R. Liao and S. Z. Li. Face Recognition Based on Multiple Facial Features. In *Proc. of the 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2000.
7. Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.
8. M. Lhuillier and L. Quan. Robust Dense Matching Using Local and Global Geometric Constraints. In *Proc. of ICPR-2000*, volume 1, pages 968–972, 2000.
9. S. Arya et al. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 45:891–923, 1998.
10. F. Samaria and A. C. Harter. Parameterisation of a Stochastic Model for Human Face Identification. In *Proc. of the 2nd IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994.
11. J. Ben-Arie and D. Nandy. A volumetric/iconic frequency domain representation for objects with application for pose invariant face recognition. *IEEE Trans. on PAMI*, 20:449–457, 1998.
12. F. Samaria. *Face Recognition Using Hidden Markov Models*. PhD thesis, University of Cambridge, 1994.
13. A. Lawrence, C. Giles, A. Tsoi, and A. Back. Face recognition: A convolutional neural network approach. *IEEE Trans. on Neural Networks*, 8(1):98–113, 1997.
14. S. H. Lin, S. Y. Kung, and L. J. Lin. Face recognition/detection by probabilistic decision-based neural network. In *Proc. of the ICASSP-96*, pages 3553–3556, 1996.
15. S. Z. Li and J. Lu. Face Recognition Using the Nearest Feature Line Method. *IEEE Trans. on Neural Networks*, 10(2):439–443, 1999.
16. G. D. Guo, S. Z. Li, and K. L. Chan. Face Recognition by Support Vector Machines. In *Proc. of the 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2000.
17. A. Nefian. *A Hidden Markov Model-Based Approach for Face Detection and Recognition*. PhD thesis, Georgia Institute of Technology, 1999.