Independent Component Analysis and Support Vector Machine for Face Feature Extraction

Gianluca Antonini, Vlad Popovici, and Jean-Philippe Thiran

Signal Processing Institute
Swiss Federal Institute of Technology Lausanne
CH-1015 Lausanne, Switzerland
{Gianluca.Antonini,Vlad.Popovici, JP.Thiran}@epfl.ch
http://ltswww.epfl.ch

Abstract. We propose Independent Component Analysis representation and Support Vector Machine classification to extract facial features in a face detection/localization context. The goal is to find a better space where project the data in order to build ten different face-feature classifiers that are robust to illumination variations and bad environment conditions. The method was tested on the BANCA database, in different scenarios: controlled conditions, degraded conditions and adverse conditions.[1].

1 Introduction

One of the most remarkable abilities of human vision is that of face detection-recognition process. Due to variations in illumination, background and facial expressions it may become complex for a computer to perform such task. Face detection-recognition algorithms are generally made up of three different steps: localization of the face region, extraction of meaningful facial features and normalization of the image respect to this features to perform the recognition step. In this paper we focus our attention on the facial feature extraction issue. Among all the possible classifi cation of the existing face detection algorithms, for our purposes we will consider the Holistic Face Models (HFM) and the Local Features Face Models (LFFM) [1].

In the HFM approach the image region containing the whole face is selected manually and a representation of the face patch is learned from examples. It is clear that the major problem with this approach is to capture all the face-class variance. Moreover, it is very diffi cult to model the geometric relationships between the different face-parts. In the LFFM approach, the basic idea is to represent the face with a set of meaningful features and not as a whole. In this way, it is easier to use any geometric information we can have about the face-class (collinear eyes positions, vertical symmetry etc...).

The feature based face detection is not a new technique. It has previously been investigated for instance in [2]. In these works they propose an implementation of the local features detectors done via the Principal Component Analysis (PCA) based classifi cation of neighbors of local maxima of the Harris corner detector.

[1] Work partially performed in the BANCA project of the IST European program with the financial support of the Swiss OFES and with the support of the IM2-NCCR of the Swiss NFS

Our approach belongs to the Local Face Feature Model category. We propose to use the Independent Component Analysis (ICA), instead of PCA, as linear transformation on the image patches and to perform the SVM classification in the ICA space. The goal is to provide a robust representation of the patches (by ICA) and train ten different classifiers (by SVM) for ten classes of features representing the face. Our algorithm can be seen as a pre-filtering stage of a face detection system.

The rest of the paper is organized as follows. Section II gives a brief introduction of ICA. Section III does the same for SVM method. Experiments and results are shown in section IV followed by some conclusions.

2 Independent Component Analysis

2.1 Overview

Assume we are observing *m* linear mixtures x_1,...,x_m of *n* independent components

$$x_j = a_{j1}s_1 + a_{j2}s_2 + ... + a_{jn}s_n \tag{1}$$

Each mixture x_j as well as each independent component s_k is a random variable. Using the vector-matrix notation we can write

$$\mathbf{x = As} \tag{2}$$

The model in Eq.1 is called independent component analysis, or ICA model. It is a *generative model*, which means that the observations are generated by a mixing process of *latent variables* which are the independent components. These variables are not directly observable and have to be estimated along with the mixing matrix A. The basic assumption in ICA is that the latent variables s_i are *statistically independent*. Technically, it means that the joint probability density is factorizable in the product of the respective marginal densities:

$$p(\mathbf{s}) = \prod_{i=0}^{N} p_i(s_i). \tag{3}$$

From the probability theory, the Central Limit Theorem tells that the distribution of a sum of independent random variables tends toward a Gaussian distribution, under certain conditions. We can use this result to assert, intuitively, that a mixture of s_i is more Gaussian distributed with respect to each of them. So, one criterion to estimate the independent components is to minimize the *gaussianity* of the s_i through some measures of *nongaussianity* like kurtosis and negentropy [3].

Another approach inspired by information theory, using the concept of differential entropy, is the minimization of mutual information

$$I(y_1, y_2, ..., y_n) = \sum_{i=0}^{n} H(y_i) - H(\mathbf{y}). \tag{4}$$

Mutual information is equivalent to the Kullback-Leibler divergence between the joint density $f(\mathbf{y})$ and the product of its marginal densities. So it represents a natural measure of the dependence between random variables and takes into account also the high-order statistics. The concepts of mutual information, negentropy and projection pursuit are all closely related [4]. Because negentropy is invariant for invertible linear transformations, finding and invertible transformation that minimizes the mutual information is roughly equivalent to find directions in which the negentropy is maximized. Again, a single direction that maximizes negentropy is a form of projection pursuit and could also be interpreted as estimation of a single component.

2.2 Why ICA?

Much of the information that perceptually distinguishes faces is contained in the higher order statistics of the images [5]. Since ICA gets more then second order statistics (covariance), it appears more appropriate with respect to PCA. The technical reason is that second-order statistics correspond to the amplitude spectrum of the image (actually, the Fourier transform of the autocorrelation function of an image corresponds to its power spectrum, the square of the amplitude spectrum). The remaining information, high-order statistics, corresponds to the phase spectrum. This is the informative part of a signal. If we remove the phase information, an image looks like noise.

3 Support Vector Machine

In this section we briefly sketch the SVM algorithm and its motivation. A more detailed description of SVM can be found in [6].

The task of learning from examples, for a two-class pattern recognition problem, can be formulated as follows: given a set of functions

$$\{f_\alpha\}_{\alpha\in\Lambda}, \quad f_\alpha : \mathbb{R} \to \{-1, +1\} \tag{5}$$

and a set of examples

$$\{(\mathbf{x}_i, y_i), i = 1, \ldots, l\} \subset \mathbb{R}^n \times \{-1, +1\}, \tag{6}$$

each one generated according to an unknown probability distribution function $P(\mathbf{x}, y)$, we want to find the function f_{α^*} which minimizes the risk of misclassification of the new patterns drawn randomly from P, given by the *risk functional*:

$$R(\alpha) = \frac{1}{2}\int |f_\alpha(\mathbf{x}) - y| \; dP(\mathbf{x}, y). \tag{7}$$

The risk functional is upper bounded by the sum of empirical risk and Vapnik-Chervonenkis (VC) confidence term (see [6]). While in practice the risk functional cannot be minimized directly, one can try to minimize its upper bound. In the case of SVM, the empirical risk is kept constant, say zero, and a minimizer for the confidence term is sought.

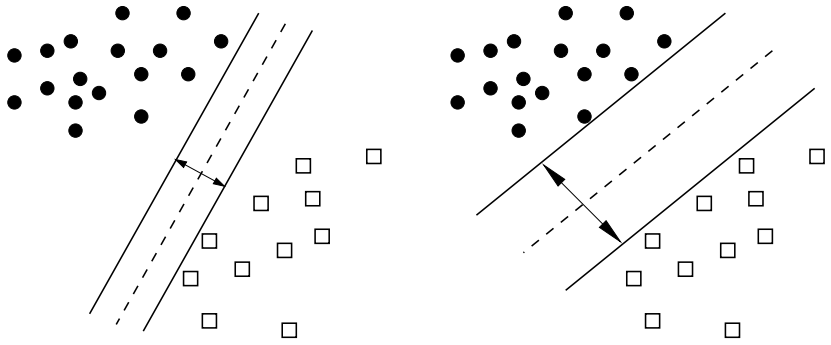
Let us consider first the simple case of linearly separable data. We are searching an *optimal separating (hyper–)plane* [2]

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \tag{8}$$

which minimizes the VC confidence term while providing the best generalization. The decision function is

$$f(\mathbf{x}) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right) \tag{9}$$

Geometrically, the problem to be solved is to find the hyperplane that maximizes the sum of distances to the closest positive and negative training examples. The distance is called *margin* (see Figure 1) and the optimal plane is obtained by maximizing $\frac{2}{\|\mathbf{w}\|}$ or, equivalently, by minimizing $\|\mathbf{w}\|^2$ subject to $y_i(\langle \mathbf{w}, \mathbf{x} \rangle + b) \geq 1$. In the case that the



(a) A possible separating plane (b) Optimal separating plane

Fig. 1. Two possible solutions for the separating plane problem. A better generalization is expected from the second case.

two classes overlap in feature space, one way to find the optimal plane is to relax the above constraints by introducing some *slack variables* ξ_i (for more details see [6]).

Introducing the Lagrange multipliers α_i, we can express the decision function as a function of them:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i \in S} y_i \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b\right) \tag{10}$$

where $S = \{i \mid \alpha_i > 0\}$, with the new constraints :

$$\sum_{i=1}^{l} y_i \alpha_i = 0 \quad \text{and} \quad \alpha_i \geq 0, \ \forall i = 1, \ldots, l \tag{11}$$

[2] We use $\langle \cdot, \cdot \rangle$ to denote the inner product operator

The vectors $\mathbf{x}_i, i \in S$ are called *support vectors* and are the only examples from the training set that affect the shape of the separating boundary.

To generalize the linear case one can project the input space into a higher–dimensional space in the hope of a better training–class separation. In the case of SVM this is achieved by using the so–called "kernel trick". In essence, it replaces the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ in (10) with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. As the data vectors are involved only in this inner products, the optimization process can be carried out in the feature space directly. Some of the most used kernel functions are:

$$\text{the polynomial kernel} \qquad K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^d \tag{12}$$

$$\text{the RBF kernel} \qquad K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2) \tag{13}$$

4 Methods and results

4.1 The BANCA database

The BANCA database is a multimodal and multi-language database. It has been recorded in 3 different scenarios: controlled, degraded and adverse. For each of the four languages (French, English, Spanish and Italian), there are 52 subjects (26 males and 26 females) each performing 12 recording sessions, with 2 recordings per session (4 sessions per scenario). In all, there are 6240 images per language. A more detailed description of the BANCA database can be found in [7].

4.2 Methods

In our experiments we have used the English subset of the BANCA database composed by 6240 images which has been divided into two equally sized subset for training and testing (in such a way that images of the same person cannot appear both in training and testing). In the test set we have taken 3120 positive examples and 3120 negative examples, so our test set size is 6240. The size of our patches is 32x32. The features we have considered are: the left corner of the left eye (P1), the central point of the left eye (P2), the right corner of the left eye (P3), the left corner of the right eye (P4), the central point of the right eye (P5), the right corner of the right eye (P6), the left and right nostrils (P7 and P8) and the left and right corners of the mouth (P9 and P10). Totally ten classes (figure 2(a)).

For each patch we perform first a PCA projection to reduce the dimensionality, passing from 1024 (32x32) components to 50 components (getting about 95% of the total variance). Starting from the PCA transformed space we apply the ICA and we train the SVM classifier in the ICA space using the radial basis function kernel. We evaluate the robustness of our classifiers in terms of *accuracy* and number of *false positive*. We use a test set composed by the manually selected feature points as positive examples and a set of random points extracted with an Harris corner detector [8] as negative examples. The results are shown in table1.

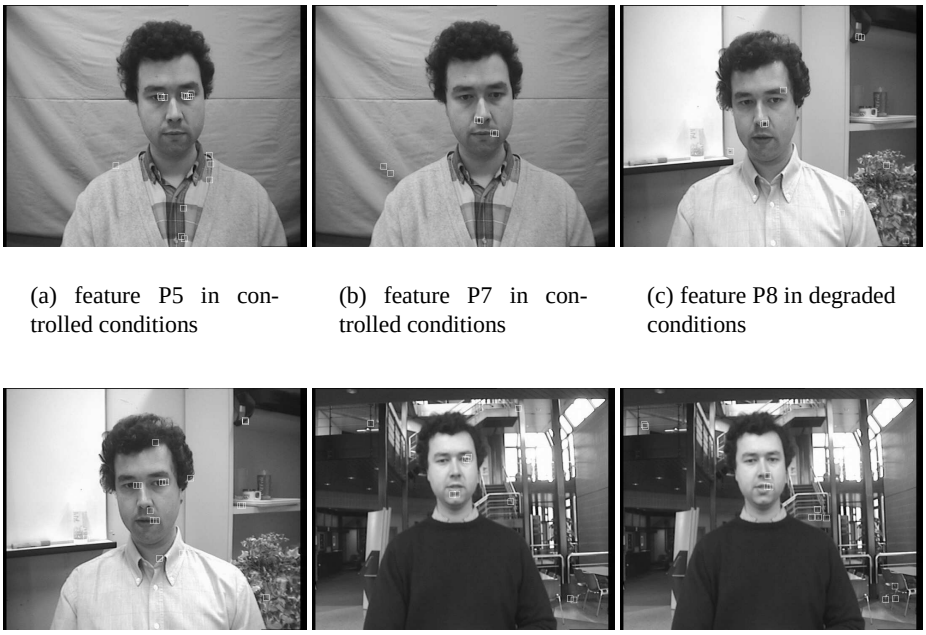
(a) The ten features (b) controlled conditions (c) degraded conditions

(d) adverse conditions

Fig. 2. The ten features (a) and the "clouds" of Harris corner in the three different conditions (b,c,d)

4.3 Results

In figures 3(a)-3(f) we show, as an example, the results we have obtained applying our models to a "cloud" of corners, from the three different scenarios, using the Harris corner detector (figures 2(b),2(c),2(d)). It is important to underline the fact that in order to avoid scanning the image in all positions, we used an Harris corner detector [8] as a prefiltering stage. It turned out that the corner detector can be tuned to pick out enough corners such that there are always corners sufficiently close to the real feature positions.



(a) feature P5 in controlled conditions

(b) feature P7 in controlled conditions

(c) feature P8 in degraded conditions

(d) feature P6 degraded conditions

(e) feature P5 in adverse conditions

(f) feature P10 in adverse conditions

Fig. 3. The features extracted from a "cloud" of corners in the three different scenarios

5 Conclusions and future work

In this work we have proposed the combined use of ICA and SVM for the facial feature extraction problem. The algorithm can be used in a face detection system as a preprocessing step, before to use other kind of information (as the geometric symmetries between the different feature positions). In order to find a better space where projects the

Features	accuracy%	false positive
P1	93.32	57
P2	93.00	37
P3	93.30	55
P4	97.01	24
P5	93.40	26
P6	92.91	26
P7	98.05	21
P8	97.80	7
P9	92.70	52
P10	95.20	30

Table 1. Numerical results on a set of 6240 test patches

data to perform the classifi cation step, would be interesting to investigate the application of some recent evolutions of the ICA as the overcomplete ICA [9] and Topographic ICA [10].

References

1. K.-K. Sung and T. Poggio. Learning human face detection in cluttered scenes. In V. Hlavac and R. Sara, editors, *Computer Analysis of Images and Patterns*, pages 432–439. Springer, Berlin,, 1995.
2. Miroslav Hamouz, Josef Kittler, Jiri Matas, and Petr Bílek. Face detection by learned affine correspondences. *LNCS*, 2396:566–575, 2002.
3. A. Hyvaerinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.
4. A. Hyvaerinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE-NN*, 10(3):626, May 1999.
5. H. M. Lades M. S. Bartlett and T. J. Sejnowski. Independent component representations for face recognition. In *Proceedings of the SPIE*, volume 3299, pages 528–539, 1998.
6. C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
7. S. Bengio, F. Bimbot, J. Mariéthoz, V. Popovici, F. Porée, E. Bailly-Baillière, G. Matas, and B. Ruiz. Experimental protocol on the BANCA database. IDIAP-RR 05, IDIAP, 2002.
8. Chris Harris and Mike Stephens. A combined corner and edge detector. *Proceedings Fourth Alvey Vision Conference*, pages 147–151, 1988.
9. Michael S. Lewicki and Terrence J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.
10. Aapo Hyvärinen, Patrik O. Hoyer, and Mika Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1527–1558, 2001.