Improved Statistical Techniques for Multi-part Face Detection and Recognition

Christian Micheloni[1], Enver Sangineto[2],
Luigi Cinque[2], and Gian Luca Foresti[1]

[1] Univeristy of Udine
Via delle Scienze 206, 33100 Udine
{michelon,foresti}@dimi.uniud.it
[2] University of Rome "Sapienza"
Via Salaria 113, 00198 Roma
{sangineto,cinque}@di.uniroma1.it

Abstract. In this paper we propose an integrated system for face detection and face recognition based on improved versions of state-of-the-art statistical learning techniques such as Boosting and LDA. Both the detection and the recognition processes are performed on facial features (e.g., the eyes, the nose, the mouth, etc) in order to improve the recognition accuracy and to exploit their statistical independence in the training phase. Experimental results on real images show the superiority of our proposed techniques with respect to the existing ones in both the detection and the recognition phase.

1 Introduction

Face recognition is one of the most studied problems in computer vision, especially w.r.t. security application. Important issues in accurate and robust face recognition is good detection of face patterns and the handling of occlusions. Detecting a face in an image can be solved by applying algorithms developed for pattern recognition tasks. In particular, the goal is to adopt training algorithms like Neural Networks [14], Support Vector Machines [1] etc. that can learn the features that mostly characterize the class of patterns to detect. Within appearance-based method, in the last years boosting algorithms [15,10] have been widely adopted to solve the face detection problem. Although they seemed to have reached a good trade-off between computational complexity and detection efficiency, there are still some considerations that leave room for further improvements in both performance and accuracy. Shapire in [13] proposed the theoretical definition of boosting. A set of *weak* hypotheses $h_1, \ldots, h_T$ is selected and linearly combined to build a more robust *strong* classifier of the form:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{1}$$

A.-B. Salberg, J.Y. Hardeberg, and R. Jenssen (Eds.): SCIA 2009, LNCS 5575, pp. 331–340, 2009.

On such an idea, the Adabost algorithm [8] proposes an efficient iterative procedure to select at each step the best weak hypothesis from an over complete set of features (e.g. Haar features). Such a result is obtained by maintaining a distribution of weights D over a set of input samples $S = \{x_i, y_i\}$ such that the error ϵ_t introduced by selecting the $t-th$ weak classifier is minimum. The error is defined as:

$$\epsilon_t \equiv Pr_{i\sim D_t}\left(h_t(x_i) \neq y_i\right) = \sum_{x_i \in S: h_t(x_i) \neq y_i} D_t(i) \tag{2}$$

where x_i is the sample pattern and y_i its class. Hence, the error introduced by selecting the hypothesis h_t is given by the sum of the current weights associated to those patterns that are misclassified by h_t. To maintain a coherent distribution D_t, that for every step t guarantees the selection of such an optimal weak classifier, the update step is as follows:

$$D_{t+i}(i) = \frac{exp\left(-y_i \sum_t h_t(x_i)\right)}{\prod_t Z_t} \tag{3}$$

where Z_t is a normalization factor that allows to maintain D as a distribution [13]. From this first formulation, new evolutions of AdaBoost have been proposed. RealBoost [9] introduced real values for weak classifiers rather then discrete ones, its development in a cascade of classifiers [16] aims to reduce the computational time for negative samples, while FloatBoost [10] introduces a backtracking mechanism for the rejection of not robust weak classifiers.

Though, all these developments suffer of a high false positive detection rate. The cause can be associated to the high asymmetry of the problem. The number of face patterns into an image is much lower than the number of non-face patterns. To balance the significance of the patterns depending on the belonging classes can be managed only by balancing the cardinality of the positives and negatives training data sets. For such a reason, the training data sets are usually composed of a larger number of negative samples than positives ones. Without this kind of control the so determined classifiers would classify positives and negatives sample in an equal way. Obviously, since we are more interested in detecting face patterns rather than non-face ones we need a mechanism that introduces a degree of asymmetry into the training process regardless the composition of the training set. Viola a Jones in [15], to reproduce the asymmetry of the face detection problem into the training mechanism, introduced a different weighting mechanism for the two classes by modifying the distribution update step. The new updating rule is the following:

$$D_{t+1}(i) = \frac{\exp\left(y_i \log\sqrt{k}\right) \exp\left(-y_i \sum_t h_t(x_i)\right)}{\prod_t Z_t} \tag{4}$$

where k is a user defined parameter that gives a different weight to the samples depending on the belonging class. If $k > 1 (< 1)$ the positive samples are considered

more (less) important, if $k = 1$ the algorithm is again the original AdaBoost. Experimentally, the authors noticed that, when determining the asymmetry parameter only at the beginning of the process, the selection of the first classifier absorbs the entire effect of the initial asymmetric weights. The asymmetry is immediately lost and the remaining rounds are entirely symmetric.

For such a reason, in this paper we propose a new learning strategy that tunes the parameter k in order to maintain active the asymmetry for the entire training process. We do that both at strong classifier learning level and at cascade definition. The resulting optimized boosting technique is exploited to train face detectors and to train other classifiers that working on face patterns can detect sub-face patterns (e.g. eyes, nose, mouth, etc.). This important features are used to achieve both a face alignment process (e.g. bringing the eyes axis horizontal) and the block extraction for recognition purposes.

Concerning the face recognition point of view, the existing approaches can be classified in three general categories [19]: *feature-based* , *holistic* and *hybrid* techniques (mixed holistic and feature-based methods). Feature based approaches extract and compares prefixed feature values from some locations on the face. The main drawback of these techniques is their dependence on an exact localization of facial features. In [3], experimental results show the superiority of holistic approaches with respect to feature based ones. On the other hand, holistic approaches consider as input the whole sub-window selected by a previous face detection step. To compress the original space for a reliable estimation of the statistical distribution, statistical "feature extraction techniques" such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) [5] are usually adopted. Good results have been obtained using Linear Discriminant Analysis (LDA)(e.g., see [18]). The LDA compression technique consists in finding a subspace T of $\mathbb{R}^M$ which maximizes the distances between the points obtained projecting the face clusters into T (where each face class corresponds to a single person). For further details, we refer to [5].

As a consequence of the limited training samples, it is usually hard to reliably learn a correct statistical distribution of the clusters in T, especially when important variability factors are present (e.g., lighting condition changes etc.). In other words, the high variance of the class pattern compared with the limited number of training samples is likely to produce an overfitting phenomenon. Moreover, the necessity of having the whole pattern as input makes it difficult to handle occluded faces. Indeed, face recognition with partial occlusions is an open problem [19] and it is usually not dealt with by holistic approaches.

In this paper we propose a "block-based" holistic technique. Facial feature detection is used to roughly estimate the position of the main facial features such as the eyes, the mouth, the nose, etc. From these positions the face pattern is split in blocks each then separately projected into a dedicated LDA space. At run time a face is partitioned in corresponding blocks and the final recognition is given by the combination of the results separately obtained from each (visible) block.

2 Multi-part Face Detection

To improve the detection rate of a boosting algorithm we considered the Asymboost technique [15] that assigns different weights to the two classes:

$$D_{t+1}(i) = \frac{\exp(y_i \log \sqrt{k}) \exp(-y_i \sum_t h_t(x_i))}{\prod_t Zt} \tag{5}$$

In particular, the idea we propose, instead of considering static the parameter k, aims to tune it on the basis of the current false positives and negatives rate.

2.1 Balancing False Positives Rate

A common way to obtain a cascade classifier with a predetermined False Positives (FP) rate $FP_{cascade}$ is to train the cascade's strong classifiers by equally spreading the FP rate among all the classifiers. This holds to the following equation:

$$FP_{cascade} = \prod_{i=1,\ldots,N} FP_{sc_i} \tag{6}$$

where FP_{sc} is the FP rate that each strong classifier of the cascade has to perform.

However, this method is not enough to allow the strong classifier to automatically control the false positive desired rate in consequence of the history of the false positives rates. In other words, if the previous level obtained a false positive rate that is under the predicted threshold, it is reasonable to suppose that the new strong classifier can consider to have a new "'smoothed"' FP threshold. For this reason, during the training of the classifier at level t we replaced FP_{sc_i} with a dynamic threshold, defined as

$$FP^{*t}_{sc_i} = FP_{sc_i} * \left(\frac{FP^{*t-1}_{sc_i}}{FP^{t-1}_{sc_i}} \right) \tag{7}$$

It is worth noticing how the false positive rate reachable by the classifier is updated at each level to obtain always a reachable rate at the end of the training process. In particular, we can see how such a value increases if at the previous step we added a weak classifier that has reduced it ($FP^{*t-1}_{sc_i} < FP^{t-1}_{sc_i}$) while decreases otherwise.

2.2 Asymmetry Control

As for the false positives rate, we can reduce the total number of false negatives by introducing a constant constraint that at each level forces the training algorithm to keep the false negatives ratio as low as possible (preferable 0). This can be achieved by balancing the asymmetry during the single strong classifier training process. The false positives-false negatives rates represent a trade-off that can be exploited to adopt a tuning strategy in the asymmetry for the two rates.

Supposing that the false negative value at the level i is quite far from the desired threshold FN_{sc_i}; at each step t of the training we can assign a different value to $k_{i,t}$, forcing the false negative ratio to decrease when $k_{i,t}$ is high (greater than one). If we suppose that the magnitude of $k_{i,t}$ directly depends on the variation of false positives obtained at step $t-1$ with respect to the desired value for such a step, we can introduce a tuning equation that increases the weight to positive samples when the false achieved positives rate is low and decreases it otherwise. Hence, for each each step $t = 1, \ldots, T$, $k_{i,t}$ is computed as

$$k_{i,t} = 1 + \frac{FP_{sc_i}^{*t-1} - FP_{sc_i}^{t-1}}{FP_{sc_i}^{*t-1}} \tag{8}$$

This equation returns a value of k that is bigger than 1 when the false positive rate obtained at the previous step has been lower than the desired one.

The Boosting technique described above have been applied both for searching the whole face and for searching some facial features. Specifically, once the face has been located in a new image (producing a candidate window D), we search in D for those candidate sub-windows representing the eyes, the mouth and the nose producing the subwindows D_{le}, D_{re}, D_m, D_n. These are used to completely partition the face pattern and produce subwindows for the forehead, the cheekbones, etc. In the next section we explain how these blocks are used for the face recognition task.

3 Block-Based Face Recognition

At training time each face image ($X^{(j)}$, $j = 1, ..., z$) of the training set is split in h independent blocks $B_i^{(j)}$ ($i = 1, ..., h$; currently $h = 9$: see Figure 1 (a)), each block corresponding to a specific facial feature. For instance, suppose that subwindow $D_m(X^{(j)})$, delimiting the mouth area found in $X^{(j)}$ is composed of the set of pixels $\{p_1, p_2, ...p_o\}$. We first normalize this window by scaling it in order to fit a window of fixed size, used for all the mouth patterns and we obtain $D'_m(X^{(j)}) = \{q_1, ...q_{M_m}\}$, where M_m is the cardinality of the standard mouth window. Block B_m, associated with D'_m is given by the concatenation of the (either gray-level or color) values of all the pixels in D'_m:

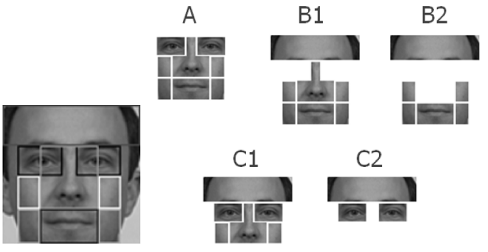
$$B_m^{(j)} = ((q_1), ...(q_{M_m}))^T. \tag{9}$$

Using $\{B_i^{(j)}\}$ ($j = 1, ..., z$) we obtain the eigenvectors corresponding to the LDA transformation associated with the i-th block:

$$W_i = (w_1^i, ..., w_{K_i}^i)^T. \tag{10}$$

Each block $B_i^{(j)}$ of each face of the gallery can then be projected by means of W_i into a subspace T_i with K_i dimensions (being $K_i << M_i$):

$$B_i^{(j)} = \mu_i + W_i C_i^{(j)}, \tag{11}$$



(a) (b)

Fig. 1. Examples of missed block tests for occlusion simulation

where μ_i is the mean value of the i-th block and $C_i^{(j)}$ is the vector of coefficients corresponding to the projection values of $B_i^{(j)}$ in T_i. We can now represent each original face $X^{(j)}$ of the gallery by means of the concatenation of the vectors $C_i^{(j)}$:

$$R(X^{(j)}) = (C_1^{(j)} \circ C_2^{(j)} \circ ... \circ C_h^{(j)})^T. \tag{12}$$

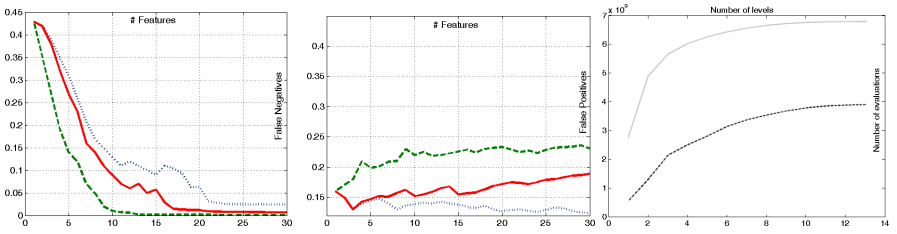
$R(X^{(j)})$ is a point in a feature space Q having $K_1 + ... + K_h$ dimensions. Note that, due to the assumed independence of block B_i from block B_j $(i \neq j)$, we can use the same image samples to separately compute both W_i and W_j. The number of necessary training samples is now dependent from the dimension of the largest block $\widehat{K} = \max_{i=1,...,h}\{K_i\}$, being $\widehat{K} < K_1 + ... + K_h$. Splitting the pattern in subpatterns offers us the possibility to deal with lower dimensional feature spaces and then using less training samples. The result is a system more robust to overfitting problems.

At testing time first of all we want to exclude from the recognition process those blocks which are not completely visible (e.g., due to occlusions). One of the problems of holistic techniques, in fact, is the necessity to consider the pattern as a whole, even when only a part of the object to be classified is visible. For this reason, at testing time we use a skin detector in order to estimate the percentage of skin in each face block and we discard from the subsequent recognition process those blocks with insufficient skin pixels. Given a test image X and a set of v visible facial blocks B_{i_l} $(l = 1, ..., v)$ of X we project each B_{i_l} into the corresponding subspace T_{i_l}, obtaining:

$$Z = (C_{i_1} \circ ... \circ C_{i_v})^T. \tag{13}$$

Z represents the *visible* patterns and is a point in the subspace U of Q. The dimensionality of U is $K_{i_1} + ... + K_{i_v}$ and U is obtained projecting Q into the dimensions corresponding to the visible blocks B_{i_l} $(l = 1, ..., v)$. Finally, we use k-Nearest Neighbor (k-NN) to search in U for the points closest to Z which indicate the gallery faces most similar to X that will be ranked and presented to the user.

It is worth noticing that the projection of Q into U is trivial and efficient to compute, since at testing time (when using k-NN) we only have to exclude,



(a) (b) (c)

Fig. 2. False positives (FP) and negatives (FN) obtained while testing small strong classifiers. The continuous, dotted and dashed lines represent performance obtained using respectively AdaBoost, AsymBoost (k=1.1) and the proposed strategy. With the same number of features, the false negatives (a) decrease faster when we apply asymmetry. Even more if we tune the asymmetry. This means our solution has a higher detection rate by using a lower number of features while keeping the false positives low (b). In (c), the lower number of features necessary by the proposed solution (dashed line) to achieve a good detection rate yields to a reduction of about 50% in computation time with respect to Adaboost (continuous line).

in computing the Euclidean distance between Z and an element $R(X^{(j)})$ of the system's database, those coefficients corresponding to the non visible blocks.

4 Experimental Results

Face Detection. The first set of experiments is aimed to compare four small single strong classifiers trained by using the presented algorithm with ones obtained by using standard boosting techniques. The input set consisted on 6500 positive (face) samples and 6500 negative (non–face) samples, collected from different sources and scaled in a standard format 27×27 pixels. In Fig. 2, the false negatives and false positive rates of three considered algorithms are plotted. The compared algorithms are AdaBoost, AsymBoost and the proposed one. Analyzing these plots we can conclude that with the same number of weak classifiers the tuning strategy that we propose achieves a faster reduction of false negatives, while keeping low false positives.

For the second experiment, two cascades of twelve levels have been trained. At each round, while the face set remains the same, a bagging process is applied to negative samples to ensure a better training of the cascade [2]. A first improvement consists in a considerable reduction of the false negatives produced by the proposed solution with respect to AsymBoost. In addition, as showed for single strong classifiers, also for cascades the number of features required by the proposed solution to achieve the same detection rate of AsymBoost is much lower. This means building a cascade with lighter strong classifiers yielding to a faster computation. As matter of fact testing both asymmetric algorithm to a benchmark test set (see Fig. 2(c)), the global evaluation costs for the proposed

solution are much lower with respect to the original AsymBoost. In particular, we have a reduction that is of about 50%.

Face Recognition. We have performed two batteries of experiments: the first with all the patterns visible (using all the facial blocks as input, i.e., with $v = h$) and the second with only a subset of the blocks. In the first type of experiments we aim to show that sub-block based LDA outperforms traditional LDA in recognizing non-occluded faces. In the second type of experiments we want to show that the proposed system is effective even with partial information, being able to correctly recognize faces with only few visible blocks.

Both types of experiments have been performed using two different datasets: the gray-scale images of the ORL [12] and (a random subset of) the colour images of the Essex [6] database. Concerning the ORL dataset, for training we have randomly chosen 5 images for each of the 40 individuals this database is composed of and we used the remaining 200 images for testing. Concerning Essex, we have randomly chosen 40 individuals of the dataset, using 5 images each for training and other 582 images of the same individuals for testing.

In the first type of experiments we have used both LDA and PCA techniques in order to provide a comparison between the two most common feature extraction techniques in both block-based and holistic recognition processes. Figure 3 shows the results concerning the top 10 corrected individuals in both the ORL and the Essex dataset. In the (easier) Essex dataset, both holistic and block-based LDA and PCA recognition techniques perform very well, with more than 98% of

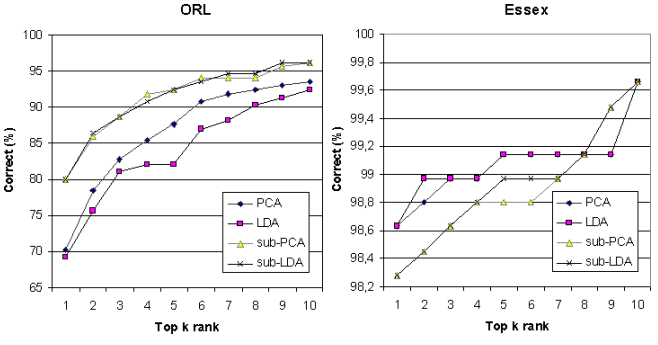


Fig. 3. Comparison between standard and sub-pattern based PCA and LDA with the ORL and the Essex datasets

Table 1. Test results obtained with missed blocks

Occlusion	ORL (%)	Essex (%)
A	71.35	93.47
B1	74.59	98.28
B2	68.11	98.45
C1	69.19	97.42
C2	62.70	96.91

correct individuals retrieved in the very first position. Traditional LDA and PCA as well as their corresponding block based versions (indicated as "sub-LDA" and "sub-PCA" respectively) have comparable results (being the difference among the four tested methods less than 1%). Conversely, in the hardier ORL dataset, sub-PCA and sub-LDA clearly outperform holistic approaches, with a difference in accuracy of about $5 - 10\%$. We think that this result is due to the fact that the lower dimensionality of each block with respect to the whole face window permits the system to more accurately learn the pattern distribution (at training time) with few training data (see Section 3).

Table 1 shows the results obtained using only subsets of the blocks. In details, we have tested the following block combinations (see Figure 1 (b)):

- A: The whole face except the forehead,
- B: The whole face except the eyes-nose zone,
- C: The whole face except the lower part.

Table 1 refers to sub-LDA technique only and to *top 1 ranking* (percentage of correct individuals retrieved in the very first position). As it is evident from the table, even with very incomplete data (e.g., the $C2$ test), block based LDA performs surprisingly well.

5 Conclusions

In this paper we have presented some improvements in state-of-the-art statistical learning techniques for face detection and recognition and we have shown an integrated system performing both tasks. Concerning the detection phase, we propose a method to balance the asymmetry of boosting techniques during the learning phase. In this way the detection performances show a faster detection and a lower FN rate. Moreover, in the recognition step, we propose to combine the results of separate classifications, each one obtained using a particular anatomically significant portion of the face. The resulting system is more robust to overfitting and can better deal with possible face occlusions.

Acknowledgments. This work was partially supported by the Italian Ministry of University and Scientific Research within the framework of the project "Ambient Intelligence: event analysis, sensor reconfiguration and multimodal interfaces"(2006-2008).

References

1. Bassiou, N., Kotropoulos, C., Kosmidis, T., Pitas, I.: Frontal face detection using support vector machines and back-propagation neural networks. In: ICIP (1), Thessaloniki, Greece, October 7–10, 2001, pp. 1026–1029 (2001)
2. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
3. Brunelli, R., Poggio, T.: Face recognition: Features versus templates. IEEE Transaction on Pattern Analysis and Machine Intelligence 15(10), 1042–1052 (1993)

4. Cristinacce, D., Cootes, T., Scott, I.: A multi-stage approach to facial feature detection. In: British Machine Vision Conference (BMVC 2004), pp. 277–286 (2004)
5. Duda, R.O., Hart, P.E., Strorck, D.G.: Pattern classification, 2nd edn. Wiley Interscience, Hoboken (2000)
6. University of Essex. The Essex Database (1994), http://cswww.essex.ac.uk/mv/allfaces/faces94.html
7. Phillips, P., Wechsler, H., Huang, J., Rauss, P.: The FERET database and evaluation procedure for face recognition algorithms. Image and Vision Computing 16(5), 295–306 (1998)
8. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: ICML, Bari, Italy, July 3–6, 1996, pp. 148–156 (1996)
9. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. The Annals of Statistics 28, 337–374 (2000)
10. Li, S.Z., Zhang, Z.: Floatboost learning and statistical face detection. IEEE Trans. Pattern Anal. Machine Intell. 26(9), 1112–1123 (2004)
11. Nefian, A., Hayes, M.: Face detection and recognition using hidden markov models. In: ICIP, Chicago, IL, USA, October 4–7, 1998, vol. 1, pp. 141–145 (1998)
12. ATeT Laboratories Cambridge. The ORL Face Database (2004), http://www.camorl.co.uk/facedatabase.html
13. Schapire, R.E.: Theoretical views of boosting and applications. In: Watanabe, O., Yokomori, T. (eds.) ALT 1999. LNCS, vol. 1720, pp. 13–25. Springer, Heidelberg (1999)
14. Smach, F., Abid, M., Atri, M., Mitéran, J.: Design of a neural networks classifier for face detection. Journal of Computer Science 2(3), 257–260 (2006)
15. Viola, P.A., Jones, M.J.: Fast and robust classification using asymmetric adaboost and a detector cascade. In: NIPS, Vancouver, British Columbia, Canada, December 3–8, 2001, pp. 1311–1318 (2001)
16. Viola, P.A., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: CVPR (1), Kauai, HI, USA, December 8–14, 2001, pp. 511–518 (2001)
17. Wiskott, L., Fellous, J.M., Malsburg, C.V.D.: Face recognition by elastic bunch graph matching. IEEE Trans. Pattern Anal. Machine Intell. 19, 775–779 (1997)
18. Xiang, C., Fan, X.A., Lee, T.H.: Face recognition using recursive fisher linear discriminant. IEEE Transactions on Image Processing 15(8), 2097–2105 (2006)
19. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. CM Computing Surveys 35(4), 399–458 (2003)