IDIAP RESEARCH REPORT

BOOSTING PIXEL-BASED CLASSIFIERS FOR FACE VERIFICATION

Yann Rodriguez [a] Sébastien Marcel [a]

IDIAP–RR 03-65

MARCH 2004

TO APPEAR IN
Biometric Authentication Workshop (BIOAW 2004) in conjunction with ECCV 2004

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail
secretariat@idiap.ch
internet
http://www.idiap.ch

[a] IDIAP, CP 592, 1920 Martigny, Switzerland

IDIAP Research Report 03-65

BOOSTING PIXEL-BASED CLASSIFIERS FOR FACE VERIFICATION

Yann Rodriguez Sébastien Marcel

MARCH 2004

TO APPEAR IN
Biometric Authentication Workshop (BIOAW 2004) in conjunction with ECCV 2004

Abstract. The performance of face verification systems has steadily improved over the last few years. State-of-the-art methods use the projection of the gray-scale face image into a Linear Discriminant subspace as input of a classifier such as Support Vector Machines or Multi-layer Perceptrons. Unfortunately, these classifiers involve thousands of parameters that are difficult to store on a smart-card for instance. Recently, boosting algorithms has emerged to boost the performance of simple (weak) classifiers by combining them iteratively. The famous AdaBoost algorithm have been proposed for object detection and applied successfully to face detection. In this paper, we investigate the use of AdaBoost for face verification to boost weak classifiers based simply on pixel values. The proposed approach is tested on a benchmark database, namely XM2VTS. Results show that boosting only hundreds of classifiers achieved near state-of-the-art results. Furthermore, the proposed approach outperforms similar work on face verification using boosting algorithms on the same database.

1 Introduction

Identity verification is a general task that has many real-life applications such as access control, transaction authentication (in telephone banking or remote credit card purchases for instance), voice mail, or secure teleworking.

The goal of an *automatic identity verification system* is to either accept or reject the identity claim made by a given person. Biometric identity verification systems are based on the characteristics of a person, such as its face, fingerprint or signature. A good introduction to identity verification can be found in [16]. Identity verification using face information is a challenging research area that was very active recently, mainly because of its natural and non-intrusive interaction with the authentication system.

The paper is structured as follow. In section 2 we first introduce the reader to the problem of face verification.

Then, we present the proposed approach, boosting pixel-based classifiers for face verification. We then compare our approach to state-of-the-results on the benchmark database XM2VTS. Finally, we analyze the results and conclude.

2 Face Verification

2.1 Problem Description

An identity verification system has to deal with two kinds of events: either the person claiming a given identity is the one who he claims to be (in which case, he is called a *client*), or he is not (in which case, he is called an *impostor*). Moreover, the system may generally take two decisions: either *accept* the *client* or *reject* him and decide he is an *impostor*.

The classical face verification process can be decomposed into several steps, namely *image acquisition* (grab the images, from a camera or a VCR, in color or gray levels), *image processing* (apply filtering algorithms in order to enhance important features and to reduce the noise), *face detection* (detect and localize an eventual face in a given image) and finally *face verification* itself, which consists in verifying if the given face corresponds to the claimed identity of the client.

In this paper, we assume (as it is often done in comparable studies, but nonetheless incorrectly) that the detection step has been performed perfectly and we thus concentrate on the last step, namely the face verification step. The problem of face verification has been addressed by different researchers and with different methods. For a complete survey and comparison of different approaches see [18].

2.2 State-of-the-art methods

The representation used to code input images in most state-of-the-art methods are often based on gray-scale face image [10, 7] or its projection into Principal Component subspace or Linear Discriminant subspace [6]. In this section, we briefly introduce one of the best method [6].

Principal Component Analysis (PCA) identifies the subspace defined by the eigenvectors of the covariance matrix of the training data. The projection of face images into the coordinate system of eigenvectors (Eigenfaces) [14] associated with nonzero eigenvalues achieves information compression, decorrelation and dimensionality reduction to facilitate decision making. A Linear Discriminant is a simple linear projection where the projection matrix is chosen according to a given criterion such as the Fisher criterion [5]. The Fisher criterion aims at maximizing the ratio of between-class scatter to within-class scatter. Finally, the Fisher Linear Discriminant subspace holds more discriminant features for classification [2] than the PCA subspace.

In [6], the projection of a face image into the system of Fisher-faces yields a representation which will emphasize the discriminatory content of the image. The main decision tool is Support Vector Machines (SVMs).

The above approach involves thousands of parameters that are difficult to store on a smart-card for instance. New approaches should be investigate to build classifiers using only hundreds of parameters. Recently, boosting algorithms has emerged to boost the performance of simple (weak) classifiers by combining them iteratively. The famous **AdaBoost** algorithm have been proposed for object detection [15] and applied successfully to face

detection [11]. AdaBoost have been applied also to face verification [1] to boost classifiers based on Haar-like features (Fig. 1) as described in [11]. Unfortunately, this boosting approach has obtained results far from the state-of-the-art.

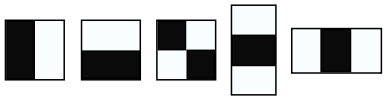

Figure 1: Five types of Haar-like features.

3 The Proposed Approach

In face verification, we are interested in particular objects, namely faces. The representation used to code input images in most state-of-the-art methods are often based on gray-scale face image. Thus, we propose to use AdaBoost to boost weak classifiers based simply on pixel values.

3.1 Feature Extraction

In a real application, the face bounding box will be provided by an accurate face detector [4], but here the bounding box is computed using manually located eyes coordinates, assuming a perfect face detection. In this paper, the face bounding box is determined using face/head anthropometry measures [3] according to a face model (Fig. 2).

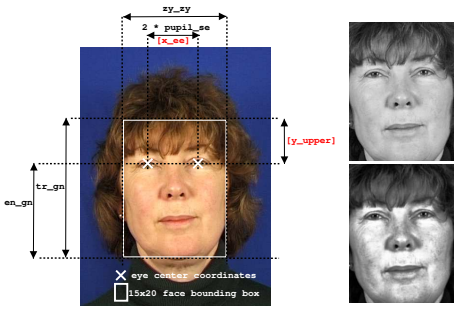

Figure 2: Face modeling and pre-processing. On the left: the face modeling using eyes center coordinates and facial anthropometry measures. On top-right: the original face image. On the bottom-right: the pre-processed face image.

The face bounding box `w`/`h` crops the physiognomical height of the face. The width `w` of the face is given by `zy_zy`/`s` where `s` = 2·`pupil_se`/`x_ee` and `x_ee` is the distance between eyes in pixels. In this model, the ratio `w`/`h` is equal to the ratio 15/20. Thus, the height `h` of the face is given by `w`·20/15 and `y_upper` = `h`·(`tr_gn` - `en_gn`) / `tr_gn`. The constants `pupil_se` (pupil-facial middle distance), `en_gn` (lower half of the craniofacial height), `tr_gn` (height of the face), and `zy_zy` (width of the face) can be found in [3].

The extracted face is downsized to a 15x20 image. Then, we perform histogram normalization to modify the contrast of the image in order to enhance important features. Finally, we smooth the enhanced image by convolving a 3x3 Gaussian ($\sigma = 0.25$) in order to reduce the noise. After enhancement and smoothing (Fig. 2), the face image becomes a feature vector of dimension 300.

3.2 Boosting Weak Classifiers

3.2.1 Introduction

A complete introduction to the theoretical basis of boosting and its applications can be found in [12]. The underlying idea of boosting is to linearly combine simple weak classifiers $h_i(x)$ to build a strong ensemble $f(x)$:

$$f(x) = \sum_{i=1}^{n} \alpha_i h_i(x)$$

Both coefficients α_i and hypothesis $h_i(x)$ are learned by the boosting algorithm. Each classifier $h_i(x)$ aims to minimize the training error on a particular distribution of the training examples.

At each iteration (i.e. for each weak classifier), the boosting procedure modifies the weight of each pattern in such a way that the misclassified samples get more weight in the next iteration. Boosting hence focuses on the examples that are hard to classify.

AdaBoost [17] is the most well known boosting procedure. It has been used in numerous empirical studies and have received considerable attention from the machine learning community in the last years. Freund et al. [17] showed two interesting properties of AdaBoost. First, the training error exponentially goes down to zero as the number of classifiers grows. Second, AdaBoost still learns after the training error reaches zero. Regarding the last point, Schapire et al. [13] shown that AdaBoost not only classifies samples correctly, but also compute hypothesis with large margins The margin of an example is defined as its signed distance to the hyperplane times its label. A positive margin means that the example is well classified. It has been shown that maximizing the margin minimizes the generalization error [13].

3.2.2 Boosting Pixel-based Weak Classifiers

We choose to boost weak classifiers based simply on pixel values, as described in [15] for face detection. The weak classifier h_i to boost is given by:

$$h_i(x) = \begin{cases} 1 & : \quad x_{f_i} \leq \theta_i \\ 0 & : \quad x_{f_i} > \theta_i \end{cases}$$

where x is the given input image, f_i is the index of the pixel to test in the image x and θ_i is a threshold. AdaBoost estimates iteratively the best feature $\{f_i, \theta_i\}$ for $1 \leq i \leq 300$.

4 The XM2VTS Database and Protocol

The XM2VTS database contains synchronized image and speech data recorded on 295 subjects during four sessions taken at one month intervals. The 295 subjects were divided, according to the *Lausanne Protocol* [9], into a set of 200 clients, 25 evaluation impostors, and 70 test impostors. Two different evaluation configurations were defined. They differ in the distribution of client training and client evaluation data. Both the training client and evaluation client data were drawn from the same recording sessions for Configuration I (LP1) which might lead to biased estimation on the evaluation set and hence poor performance on the test set. For Configuration II (LP2) on the other hand, the evaluation client and test client sets are drawn from different recording sessions which might lead to more realistic results. This led to the following statistics:

- Training client accesses: 3 for LP1 and 4 for LP2
- Evaluation client accesses: 600 for LP1 and 400 for LP2
- Evaluation impostor accesses: $40,000$ (25 * 8 * 200)
- Test client accesses: 400 (200 * 2)
- Test impostor accesses: $112,000$ (70 * 8 * 200)

Thus, the system may make two types of errors: *false acceptances* (FA), when the system accepts an *impostor*, and *false rejections* (FR), when the system rejects a *client*. In order to be independent on the specific dataset distribution, the performance of the system is often measured in terms of these two different errors, as follows:

$$\text{FAR} = \frac{\text{number of FAs}}{\text{number of impostor accesses}} , \tag{1}$$

$$\text{FRR} = \frac{\text{number of FRs}}{\text{number of client accesses}} . \tag{2}$$

A unique measure often used combines these two ratios into the so-called *Half Total Error Rate* (HTER) as follows:

$$\text{HTER} = \frac{\text{FAR} + \text{FRR}}{2} . \tag{3}$$

Most verification systems output a score for each access. Selecting a threshold over which scores are considered genuine clients instead of impostors can greatly modify the relative performance of FAR and FRR. A typical threshold chosen is the one that reaches the *Equal Error Rate* (EER) where FAR=FRR on a separate validation set.

5 Experimental Results

In this section, we provide experimental[1] results obtained by our approach, pixel-based boosted weak classifiers, on the configuration I of the Lausanne Protocol. We compare the results obtained to the state-of-the-art and to similar work using AdaBoost.

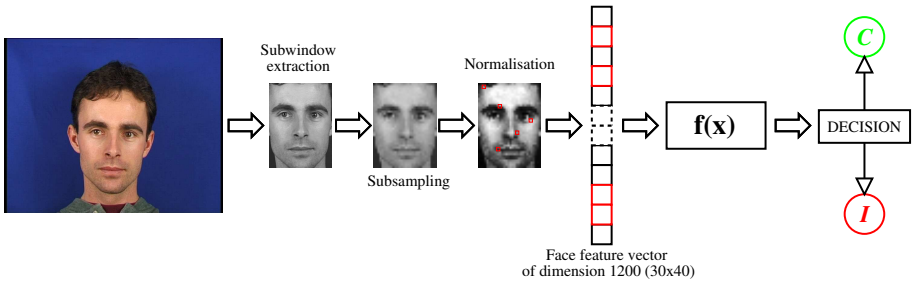

Figure 3: Pixel-based boosted classifier for face verification.

For each client, three shots are available. Each shot was slightly shifted, scaled and mirrored to obtain 220 examples. 2x220 patterns were used for training the client model and 1x220 patterns were used as a validation set to evaluate a threshold decision. The negative samples (pseudo-impostors) were generated by taking the three original shots of all other clients ((200-1) clients x 3 shots = 1194 patterns). A model has been trained for each client.

In table 1, we provide the results obtained by our boosting approach (*AdaPix*) using different number of classifiers (50, 100, 150, 200). We provide also results obtained by a state-of-the-art approach, namely Normalized Correlation (*NC*) [8], and results obtained using boosted classifiers based on seven Haar-like features [1] (*AdaHaar7*).

From these results, it can be shown that the performance of *AdaPix* increase when increasing the number of classifiers. It can be shown also that they can be compared to the state-of-the-art (*NC*). *AdaPix* outperforms *AdaHaar7* with less classifiers. Furthermore, *AdaHaar7* obtained results far from the state-of-the-art. As a fair comparison, we used our AdaBoost algorithm to boost weak classifiers for the three first types (Fig. 1) of Haar-like features (*AdaHaar3*), and we obtained an HTER two times smaller than *AdaHaar7* with two times less classifiers.

[1] The machine learning library used for all experiments is Torch *http://www.torch.ch*.

Model	FAR	FRR	HTER
NC [8]	3.46	2.75	3.1
AdaHaar7 200 [1]	6.9	8.8	7.85
AdaPix 50	3.34	4.0	3.67
AdaPix 100	3.16	3.5	3.33
AdaPix 150	3.11	3.5	3.30
AdaPix 200	2.75	3.0	**2.87**
AdaHaar3 100	2.29	5.0	3.64

Table 1: Comparative results in terms of FAR/FRR and HTER on the configuration I

6 Conclusion

In this paper, we proposed the use of AdaBoost for face verification to boost weak classifiers based simply on pixel values. The proposed approach was tested on a benchmark database, namely XM2VTS, using its associate protocol. Results have shown that boosting only hundreds of classifiers achieved near state-of-the-art results. Furthermore, the proposed approach outperforms similar work on face verification using boosting algorithms on the same database.

Boosting algorithms will certainly be used more and more often in face verification. A new direction will be probably, to combine the efficiency of boosting algorithms with discriminant features such as LDA.

Acknowledgments

The author wants to thank the Swiss National Science Foundation for supporting this work through the National Center of Competence in Research (NCCR) on "Interactive Multimodal Information Management (IM2)".

References

[1] B. Fröba, S. Stecher and C. Kübleck. Boosting a Haar-Like Feature Set for Face Verification. In *Proceedings of Audio and Video based Person Authentication*, pages 617–624, 2003.

[2] P. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. In *ECCV'96*, pages 45–58, 1996. Cambridge, United Kingdom.

[3] L.G. Farkas. *Anthropometry of the Head and Face*. Raven Press, 1994.

[4] R. Féraud, O. Bernier, J-E. Viallet, and M. Collobert. A fast and accurate face detector based on Neural Networks. *Transactions on Pattern Analysis and Machine Intelligence*, 23(1), 2001.

[5] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(II):179–188, 1936.

[6] K. Jonsson, J. Matas, J. Kittler, and Y.P. Li. Learning support vectors for face verification and recognition. In *4th International Conference on Automatic Face and Gesture Recognition*, pages 208–213, 2000.

[7] A. Kostin, M. Sadeghi, J. Kittler, and K. Messer. On representation spaces for SVM based face verification. In *Proceedings of the COST275 Workshop on The Advent of Biometrics on the Internet*, Rome, Italy, 2002.

[8] Y. Li, J. Kittler, and J. Matas. On matching scores of LDA-based face verification. In T. Pridmore and D. Elliman, editors, *Proceedings of the British Machine Vision Conference BMVC2000*. British Machine Vision Association, 2000.

[9] J Lüttin. Evaluation protocol for the the XM2FDB database (lausanne protocol). Technical Report COM-05, IDIAP, 1998.

[10] S. Marcel and S. Bengio. Improving face verification using skin color information. In *Proceedings of the 16th ICPR*. IEEE Computer Society Press, 2002.

[11] P. Viola and M. Jones. Robust Real-time Object Detection. In *IEEE ICCV Workshop on Statistical and Computational Theories of Vision*, 2001.

[12] R. Meir and G. Rätsch. *An introduction to Boosting and Leveraging*. 2003.

[13] R.E. Schapire, Y. Freund, P.L. Bartlett, and W.S. Lee. "Boosting the margin: a new explanation to the effectiveness of voting methods". *Annals of statistics*, 26(5), 1998.

[14] M. Turk and A. Pentland. Eigenface for recognition. *Journal of Cognitive Neuro-science*, 3(1):70–86, 1991.

[15] V. Pavlovic and A. Garg. Efficient Detection of Objects and Attributes Using Boosting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[16] P. Verlinde, G. Chollet, and M. Acheroy. Multi-modal identity verification using expert fusion. *Information Fusion*, 1:17–33, 2000.

[17] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the IEEE International Conference on Machine Learning*, pages 148–156, 1996.

[18] J. Zhang, Y. Yan, and M. Lades. Face recognition: Eigenfaces, Elastic Matching, and Neural Nets. In *Proceedings of IEEE*, volume 85, pages 1422–1435, 1997.