A Comparison of Face/Non-Face Classifiers

Erik Hjelmås[1,2] and Ivar Farup[2]

[1] Dept. of Informatics, Univ. of Oslo, P. O. Box 1080 Blindern, N-0316 Oslo, Norway
[2] Faculty of Technology, Gjøvik College, P. O. Box 191, N-2801 Gjøvik, Norway
{erikh,ivarf}@hig.no

Abstract. Most face detection algorithms can be divided into two sub-problems, initial visual guidance and face/non-face classification. In this paper we propose an evaluation protocol for face/non-face classification and provide experimental comparison of six algorithms. The overall best performing algorithms are the baseline template matching algorithms. Our results emphasize the importance of preprocessing.

1 Introduction

Face detection is an important and necessary first step in most face recognition applications. Face detection serves to localize potential face regions in images and classify them as faces or non-faces. This is a difficult task due to the dynamic appearance and variability of faces as opposed to more static objects such as vehicles or weapons. In addition to face recognition, areas such as content-based image retrieval, intelligent human-computer interfaces, crowd surveillance, video coding and email content security also make use of face detection algorithms.

The last decade has shown a great deal of research effort put into face detection technology. A comprehensive survey can be found in Hjelmås and Low [4], where the algorithms are classified as feature-based or image-based. However, not much work has been done on comparing existing algorithms. Some of the image-based algorithms report results on a common dataset (the CMU/MIT dataset), but there does not exist a specific evaluation protocol. This has lead to different interpretations of testing parameters for this set, which makes it hard to compare the algorithms.

In this paper we provide an experimental comparison of six face detection algorithms, categorized as two baseline, two image-based and two feature-based algorithms. One of the feature-based algorithms is a new version of an existing technique, while the rest are implemented based on previously published papers by other authors. The algorithms are selected based on findings in [4], and also to represent significantly different approaches. We also propose an evaluation protocol for the face/non-face classifier in face detection algorithms.

In section 2, we present an overview of the dataset we have selected for training and testing, while section 3 describes the testing protocol in detail. Section 4 briefly presents the algorithms (since they are described in more detail elsewhere), section 5 contains the experimental results and discussion.

2 The dataset

The dataset consists of images from the XM2VTS [7] and AR [6] face databases, and non-face images collected from the world wide web. The XM2VTS dataset is used for training. It contains 8 images of 295 subjects for a total 2360 images. All images are frontal view face images with a high degree of variation with respect to skin color, hair style, facial hair and glasses. The images are taken at four sessions with a month interval between sessions. For this training set, the coordinates of the eyes are available. For testing, we use the AR dataset with 3313 images from 136 subjects where most of the subjects images have been captured during two sessions with a 2 week interval between the sessions, from which we define the following subsets:

Easy An easy dataset with 1783 face images. All subjects vary their facial expression, and there are large variations in lighting, but there are no facial occlusions. In 14% of the images the subjects where told to scream when the image was captured, thus these images have an extreme facial expression.

Sunglasses A difficult dataset with 765 face images. All subjects are wearing dark sunglasses.

Scarf A difficult dataset with 765 face images. All subjects are wearing a scarf covering the mouth area.

From the world wide web, we have collected manually a set of 67 large images with considerable structure, which might contain face-like patterns, which we use as the negative test set. In addition we have further collected a few large images for bootstrap training of the SNoW algorithm (described later).

The resolution of the training images (XM2VTS) are originally 720×576, but we only use an extracted window covering the center of the face (rescaled to 20×20 or 60 × 60 pixels, and geometrically normalized with respect to the eyes). Similarly, the resolution of testing images (AR) are originally 768×576, but we focus the search on subset covering the facial area (see the following section for details). The test sets and training sets are non-overlapping. All images are converted to 8 bit grayscale images (256 graylevels).

3 The evaluation protocol

Most face detection tasks can be divided into two steps, where the first step is an algorithm for visual guidance or simply an exhaustive search, and the second step is the actual face/non-face classification. In this section we propose a protocol for evaluating the second step. Not all proposed face detection algorithms work in this two-step fashion, but since the general problem of face detection can be decomposed into these two steps, all face detection approaches would benefit (in terms of accuracy) from decomposing or combining their algorithm this way. Decomposing the problem leads to easier selection of the appropriate technique for the two sub-problems. The key elements of the evaluation protocol are the following (tailored to the datasets used in our experiments):

- The face classifiers generate a confidence score s_{algo} where $algo \in \{\mathbf{bE}, \mathbf{bC}, \mathbf{PCA}, \mathbf{SNoW}, \mathbf{Gradient}, \mathbf{Gabor}\}$ indicates the face classifier algorithm.
- The multiresolution scanning algorithm: a $n \times n$ window ω scans the entire image with 1 pixel step size and the image is subsampled by a factor of 1.2 until all scales and locations have been included. The face classifiers are applied at each location and scale. n is set to 20 for the image-based and baseline algorithms, and 60 for the feature-based algorithms. However, the 20×20 windows are just downscaled versions of the 60×60 windows in order to have the same number of testing windows ω for all algorithms.
- A correct detection of a face in a face image I is registered if the window ω which produces the highest confidence score ($\max_\omega(s_{algo})$) is *correctly centered* in I. We have manually located the center (x_c, y_c) of the face for all the test images, so we define ω correctly centered to be ω located such that its center region $\{(\frac{n}{2} \pm \frac{n}{4}, \frac{n}{2} \pm \frac{n}{4})\}$ encompasses (x_c, y_c).
- The correct face detection rate CD is simply

$$CD_{testset} = \frac{\text{number of images with face correctly detected}}{\text{total number of images}}$$

  where $testset \in \{\mathbf{Easy}, \mathbf{Sunglasses}, \mathbf{Scarf}\}$ indicates the test set used, and the total number of images is 1783 for the **Easy** dataset and 765 for the **Scarf** and **Sunglasses** dataset. We know that for the face images there is only one face present in each image.
- For the false alarm rate FA, we are simply interested in the number of false alarms relative to the total number of windows ω produced by the multiresolution scanning algorithm on the negative test set. This number is 5938360, so the false alarm rate is computed from

$$FA = \frac{\text{number of false detections}}{5938360}$$

  A false alarm is a window ω where the face classifier produces a $s_{algo} > t_{algo}$. We do not count false alarms in the face images (the positive test sets).
- Results are reported in terms of ROC (Receiver Operator Characteristics) curves, which shows the trade-off between correct face detection rate CD and the false alarm rate FA. The threshold t_{algo} for the face classifier is varied in a range to produce a false alarm rate $10^{-4} \leq FA \leq 10^{-1}$.

4 The algorithms

Baseline template algorithms Standard template matching is used as baseline algorithms for comparison. The training images are geometrically normalized such that a 20×20 window encompassing eyes in fixed positions, nose and mouth, can be extracted. We compute the template by simply averaging these training images. Matching is performed by measuring either Euclidean distance (**bE**) or computing the normalized correlation coefficient (**bC**) between template and testing window.

Image-based: PCA algorithm Principal Component Analysis (PCA) can be used to create a face space consisting of eigenfaces as an orthogonal basis [11], on which new faces can be projected to achieve a more compact representation. In our implementation, we use the reconstruction error $\epsilon^2 = \|\tilde{\omega}\| - \sum_{i=1}^{n} y_i^2$ (where y_i are projection coefficients and $\|\tilde{\omega}\|$ is the mean subtracted window) as a measure for the score s_{pca}. We only keep the first principal component for representation (thus $n = 1$).

Image-based: SNoW algorithm The SNoW (Sparse Network of Winnows) learning architecture, proposed by Roth in [1], has been successfully applied to face detection by Roth et al. in [10]. We have implemented this algorithm using the software available from the website of Roth for training, and our own implementation for testing. The technical details of of the algorithm are described in [10]. We also use a training procedure similar to the bootstrap training proposed by Sung and Poggio [13].

Feature-based: Gradient algorithm This algorithm is a variant of the work of Maio and Maltoni [5]. From a 60×60 window a directional image consisting of 20×20 pairs of directions and weights is extracted using the algorithm by Donahue and Rokhlin [3]. This is compared with a constructed template representing the characteristic features of a face using the distance function from [5]. In contrast with the original work of Maio and Maltoni, the constructed template does not contain the ellipsis outlining the face, and the distances between the facial elements in the constructed template are chosen to resemble the template used in the baseline algorithms as closely as possible.

Feature-based: Gabor algorithm Gabor features are widely applied for local feature extraction in face recognition systems and have also been used for face detection [9] and facial feature detection [12]. Gabor features are extracted using a set of 2D Gabor filters [2]. In our implementation we use a set of 40 filters (5 sizes, 8 orientations) generated by a wavelet expansion. We create a Gabor template which is a 60×60 window where the set of 40 Gabor coefficients have been extracted at the two locations corresponding to the eyes. In other words, we have a template which simply represents the average eyes. We only keep the magnitude of the complex coefficients and compare the template with the extracted subwindow at each location using the normalized correlation coefficient.

5 Results and discussion

We try out several combinations of two preprocessing techniques – subtraction of best fit linear plane and histogram equalization – using the **bE**-algorithm. Figure 1A shows that the preprocessing is of major importance for the algorithm to work correctly. The best CD is obtained when both kinds of preprocessing are applied to both the test images and the template. This combination is thus applied for the remaining algorithms (except for the Gabor algorithm which is not as dependent on preprocessing).

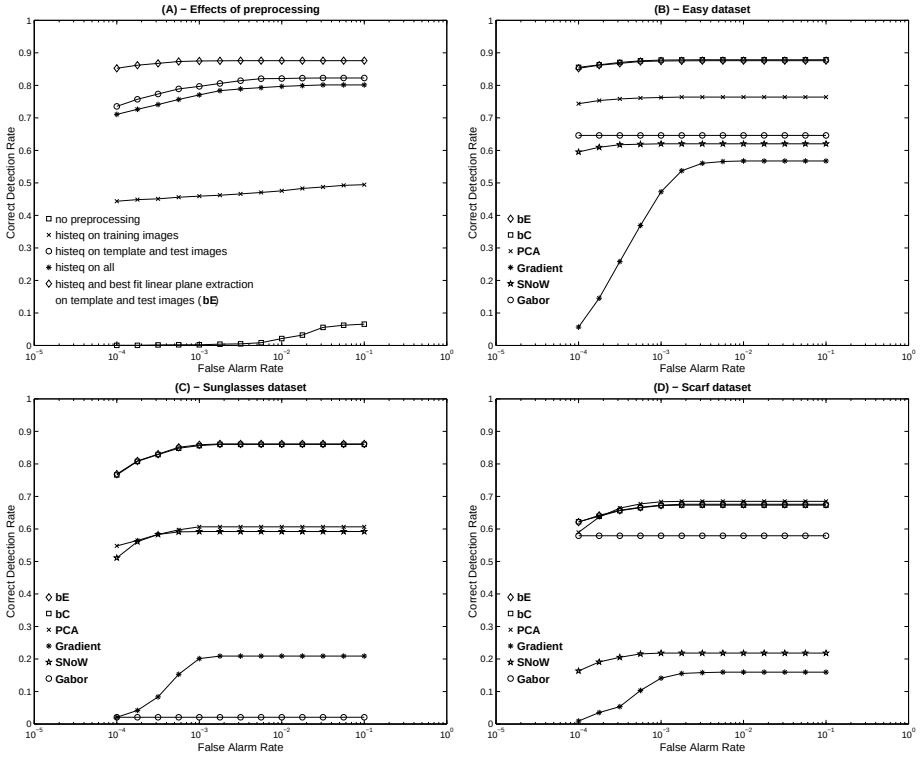

Fig. 1. Experimental results.

The results for all algorithms are shown in figure 1B for the **easy** dataset, figure 1C for the **sunglasses** dataset, and figure 1D for the **scarf** dataset. The baseline template matching algorithms are the overall best performing algorithms.

The **PCA** algorithm gives the best results when using only the first principal component, thus reducing the algorithm to a modified correlation measure. The reason for this is possibly that the size of the training set is not large enough to provide a general basis for representing the class of faces. We believe that this could be the reason since a general face class consisting of geometrically normalized faces should be Gaussian [8], and examination of the training data when plotting the projection coefficients of the first two principal components showed us that this is not the case.

The size of the training set is possibly also the reason to the poor performance of the **SNoW** classifier, since the classifier had no problems learning the face/non-face classification during training and initial testing.

The abandoning of the ellipsis around the face introduced an important alteration for the **Gradient** algorithm compared to the original work of Maio and Maltoni [5]. This might explain why the algorithm performs less than ideally.

In the original work, the total weight of the ellipsis in the distance function was approximately 2–3 times the weight of the remaining template, indicating the importance of the ellipses.

Selection of the Gabor filters for the **Gabor** algorithm was accomplished by manual inspection, and we have no reason to believe that these filters are optimal for representing the face class (in terms of the eyes here).

To our knowledge, detailed comparison of the preprocessing effects in face detection has not been presented earlier, thus figure 1A is quite significant. Simple template matching algorithms are not always used as a baseline for comparison, and our results should be taken as a strong indication that this is necessary. Due to the complexity of the other algorithms such as different selection of training set size, training parameters, template and filter design, improved performance can most likely be achieved. However, in our scenario, the simple baseline algorithms show impressive performance with the right kind of preprocessing.

References

[1] A. J. Carlson, C. M. Cumby, J. L. Rosen, and D. Roth. SNoW user's guide. Technical Report UIUC-DCS-R-99-210, UIUC CS dept, 1999.

[2] J. G. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20:847–856, 1980.

[3] M. J. Donahue and S. I. Rokhlin. On the use of level curves in image analysis. *Image Understanding*, 57:185–203, 1993.

[4] E. Hjelmås and B. K. Low. Face detection: A survey. submitted.

[5] D. Maio and D. Maltoni. Real-time face location on gray-scale static images. *Pattern Recognition*, 33:1525–1539, 2000.

[6] A. M. Martinez and R. Benavente. The AR face database. Technical Report CVC 24, School of Elec. and Comp. Eng., Purdue University, 1998.

[7] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In *Second International Conference on Audio and Video-based Biometric Person Authentication*, 1999.

[8] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1), 1997.

[9] K. Okada, J. Steffens, T. Maurer, H. Hong, E. Elagin, H. Neven, and C. v. d. Malsburg. The Bochum/USC face recognition system and how it fared in the FERET phase III test. In *Face Recognition: From Theory to Application*. Springer, 1998.

[10] D. Roth, M.-H. Yang, and N. Ahuja. A SNoW-based face detector. In *Advances in Neural Information Processing Systems 12 (NIPS 12)*. MIT press, 2000.

[11] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America*, 4:519–524, 1987.

[12] F. Smeraldi, O. Carmona, and J. Bigün. Saccadic search with Gabor features applied to eye detection and real-time head tracking. *Image and Vision Computing*, 18:323–329, 2000.

[13] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, January 1998.