Face Recognition Using Boosted Local Features

Jones, M.; Viola, P.

Abstract

This paper presents a new method for face recognition which learns a face similarity measure from example image pairs. A set of computationally efficient "rectangle" features are described which act on pairs of input images. The features compare regions within the input images at different locations, scales, and orientations. The AdaBoost algorithm is used to train the face similarity function by selecting features. Given a large face database, the set of face pairs is too large for effective training. We present a sampling procedure which selects a training subset based on the AdaBoost example weights. Finally, we show state of the art results on the FERET set of faces as well as a more challenging set of faces collected at our lab.

Face Recognition Using Boosted Local Features

Michael J. Jones

Mitsubishi Electric Research Laboratory
201 Broadway
Cambridge, MA 02139

Paul Viola

Microsoft Research
One Microsoft Way
Redmond, WA 98052

Abstract

This paper presents a new method for face recognition which learns a face similarity measure from example image pairs. A set of computationally efficient "rectangle" features are described which act on pairs of input images. The features compare regions within the input images at different locations, scales, and orientations. The AdaBoost algorithm is used to train the face similarity function by selecting features. Given a large face database, the set of face pairs is too large for effective training. We present a sampling procedure which selects a training subset based on the AdaBoost example weights. Finally, we show state of the art results on the FERET set of faces as well as a more challenging set of faces collected at our lab.

1 Introduction

This paper presents a new method for face recognition which combines some of the properties of previous approaches, while introducing a new set of image feautures and a new learning algorithm. The recognition results on the only publically available database which has been widely evaluated (FERET) match the best published results. Evaluation on a more realistic internal database are very promising.

One widely influential face recognition algorithm is that of Moghaddam and Pentland [3]. They use a statistical approach which learns which types of variations are observed in different images of the same individual (intra-personal variation). This is compared to the distribution of extra-personal variation. The intra-personal and extra-personal distributions estimated are assumed to be Gaussian, and are approximated with principal eigenvectors which are global. One key advantage of this approach is that learning is used to focus on important difference between individuals.

Another widely influential algorithm is that produced by von der Malsburg and colleagues [13, 4]. They use a set of complex image features they call Gabor Jets. A large set of these features evaluated on gallery images are then registered to probe images using elastic matching. One key advantage of this approach is that it uses complex local measures of faces. This allows the technique to pick up the fact that an individual's eyes or nose is very recognizable. Since this approach does not use learning it is immediately applicable to new types of images. However, it cannot learn to ignore intrapersonal variation and accentuate extrapersonal variation.

Like the Moghaddam-Pentland approach, our system learns to distinguish between intrapersonal and extrapersonal variation using a large database of images. Like the von der Malsburg approach our approach compares faces using a combination of local features. But, unlike the work of von der Malsburg's group, the scale, orientation, and form of each feature is learned. A small set of features are selected, from an extremely large set, because they are sensitive to extrapersonal variations, and insensitive to intrapersonal variation. Unlike the work of Moghaddam and Pentland no distributional assumptions are made about the intrapersonal or extrapersonal space. The final classifier, which combines a few hundred localized features can be evaluated in less than a millisecond on a pair of input images.

The overall approach is the construction of a face similarity function which is evaluated on two face images (that have been cropped and rectified to normalize for translation, scale and rotation). This function is learned from a database which includes examples of "same face" and "different face" image pairs. This similarity can be thresholded to yield a binary decision of same/different, or it can be used to find the most similar face in a gallery. The face similarity function consists of a linear combination of "rectangle" features not unlike those described by Viola and Jones [11, 12]. These features have been modified to apply to a pair of input images, rather than a single image. This modification will be described in detail in section 2.

Given a database that includes N images of each of K individuals, the total number of pairs is $\binom{KN}{2}$. A small minority, $K\binom{N}{2}$, of these pairs display the same individual. Any approach for learning the similarity function must

explicitly handle both the overwhelmingly large number of pairs, and the grossly unequal distribution of positive and negative examples. We describe an extension of AdaBoost which samples a small subset of examples from the total, which nevertheless will converge to the optimal classifier for the entire set (see Section 3).

Finally we present results in section 4 on both the FERET set of faces and a set of faces we have collected at our lab.

2   Filters, Features and Classifiers

The general problem of face recognition is actually comprised of two related problems: recognition and verification. Both problems assume a **gallery** of face images with known identities. In the recognition problem, a **probe** face is presented to the system and the task is to determine if the probe face belongs to any of the people in the gallery. In the verification problem, a probe face is presented along with its alleged identity. The problem is to compare the probe face with the gallery face corresponding to the alleged identity and determine if they are the same person.

Both of these problems can be solved with a face similarity function. The face similarity function takes two face images as input and outputs a measure of their similarity:

$$F(I_1, I_2) \in \mathcal{R} \tag{1}$$

where I_1 and I_2 are the input face images which have been cropped and rectified as described later. The similarity measure can be thresholded to yield a verification system. Alternatively by selecting the most similar image in a gallery, $F()$ can be used as a recognition algorithm.

In this paper the face similarity function is a sum of features f_i:

$$F(I_1, I_2) = \sum_{i=1}^{N} f_i(I_1, I_2). \tag{2}$$

A feature consists of a filter which acts on both input images:

$$f_i(I_1, I_2) = \begin{cases} \alpha & \text{if } |\phi_i(I_1) - \phi_i(I_2)| > t_i \\ \beta & \text{otherwise} \end{cases} \tag{3}$$

where $t_i \in \mathcal{R}$ is a feature threshold and ϕ_i we call a "filter", because it is a scalar function of the image. We adopt a set of linear filters similar to those used for object detection [5, 11]. The set of rectangle filters used in this work is shown in figure 1. With the addition of a "diagonal" filter these are a superset of those used by Viola and Jones

Rectangle filters are computed by summing the intensities of all pixels in the dark regions and subtracting the sum of the intensities of all pixels in the light regions. For filters
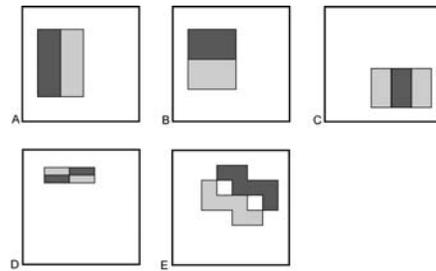

Figure 1: A single example of each type of rectangle filter. Note, the complete set of filters ranges over all scales, aspect rations and locations in the analysis window.

with more dark regions than light (or vice versa), a multiplier is factored in to make the total number of pixels in dark rectangles the same as the total number of pixels in light rectangles. The computation of rectangle filters can be greatly sped up by first computing an integral image representation of the input images [11, 1]. With the integral image representation, computation of a rectangle filter can be performed in constant time (i.e. independently of the number of pixels).

Our intuition regarding faces is that isolated structures such as eye brows, nose, or lips, provide valuable information for recognition.[1] Each of these structures undergoes a set of allowable intrapersonal variations. Algorithmic face recognition requires features which measure face structures directly and robustly, and a learning algorithm that can discover the difference between intrapersonal and extrapersonal variation.

Each feature defined above measures a particular property, at a given location, scale, and aspect ratio, and is assigned a weight. The filters themselves pick up characteristics of a face such as lines, edges and dark regions adjacent to light regions. The feature thresholds determine which variations are acceptable, and which are unacceptable. If a region of the face, such as the hair, is not a good indicator of face similarity in the training images then it is likely that no filter will be chosen in this region by the learning algorithm. Thus the learning algorithm will ignore such non-informative regions of the input.

Currently, the features we use are equivalent to first computing a difference image and then computing a filter on that difference image (followed by the absolute value and a threshold). However, the formulation we use is easily generalizable to the case of different filters being computed in the two images or a nonlinear filter being computed in each image. In these cases, a difference image could not be used.

This brings us to the problem of learning the best set of features for distinguishing same faces from different faces.

[1]See the work of Penev and Atick for similar insights [6]

2

The learning problem can be stated as follows. Given a large library of rectangle filters and a set of positive and negative examples, find the best filters, thresholds and weights (α and β) for creating a classifier that separates the positive examples (same face pairs) from the negative examples (different face pairs). The learning problem is solved using AdaBoost and an extension to AdaBoost that allows us to use very large training sets.

3 Learning algorithm

We use an improved version of AdaBoost, derived from the work of Schapire and Singer [10] that uses confidence-rated predictions. This version of AdaBoost simplifies notation and allows a simplified analysis. Also, in order to handle very large datasets we modify the AdaBoost algorithm by using the concept of resampling. Resampling is explained in section 3.2.

3.1 AdaBoost

Following Schapire and Singer [10], the AdaBoost algorithm assigns to each example x_i a weight D_i. D_i is initialized to $1/N$ where N is the total number of examples. The correct label for each example is $y_i \in \{+1, -1\}$.

AdaBoost proceeds in rounds. On each round a weak classifier is chosen. The only requirement of the weak classifier is that it has an error rate less than 0.5. In our case, our input vectors are image pairs ($x_i = (I_1^i, I_2^i)$) and a weak classifier consists of a single feature which contains a rectangle filter acting on each input image in the pair. Let

$$h_j(I_1, I_2) = |\phi_j(I_1^i) - \phi_j(I_2^i)| \qquad (4)$$

then

$$f_j(x_i) = f_j(I_1^i, I_2^i) = \begin{cases} \alpha & \text{if } h_j(I_1^i, I_2^i) > t_j \\ \beta & \text{otherwise} \end{cases} \qquad (5)$$

.

On each round of AdaBoost, we choose the weak classifier to be the ϕ_j and t_j for which

$$\epsilon_j = \sum_{\substack{i \,:\, y_i = +1 \\ \wedge\, h(x_i) \le t}} D_i \quad + \quad \sum_{\substack{i \,:\, y_i = -1 \\ \wedge\, h(x_i) > t}} D_i \qquad (6)$$

is minimized. The first term is the sum of the weights of the examples that are false negatives of $f_j(x)$ and the second term is the sum of the weights of the examples that are false positives. So minimizing the sum of these terms minimizes the weighted error.

Once the optimal filter and threshold are found by minimizing the weighted error, good values for α and β must be

computed. Schapire and Singer show that minimizing

$$Z = \sum_{i=1}^{N} D_i e^{-y_i f(x_i)} \qquad (7)$$

is a good criteria for choosing weak hypotheses. This criteria can be used to compute good values for α and β as follows.

First split the sum into a sum over negative and positive examples:

$$Z = \sum_{i:y_i=+1} D_i e^{-f(x_i)} + \sum_{i:y_i=-1} D_i e^{f(x_i)}$$

$$= \sum_{\substack{i \,:\, y_i = +1 \\ \wedge\, h(x_i) > t}} D_i e^{-\alpha} \quad + \quad \sum_{\substack{i \,:\, y_i = +1 \\ \wedge\, h(x_i) \le t}} D_i e^{-\beta} \quad +$$

$$\sum_{\substack{i \,:\, y_i = -1 \\ \wedge\, h(x_i) > t}} D_i e^{\alpha} \quad + \quad \sum_{\substack{i \,:\, y_i = -1 \\ \wedge\, h(x_i) \le t}} D_i e^{\beta}$$

Let W_+^- equal the total weight of the positive examples that are labeled negative (false negatives).

$$W_+^- = \sum_{\substack{i \,:\, y_i = +1 \\ \wedge\, h(x_i) \le t}} D_i$$

Similarly, let

$$W_-^+ = \sum_{\substack{i \,:\, y_i = -1 \\ \wedge\, h(x_i) > t}} D_i$$

$$W_+^+ = \sum_{\substack{i \,:\, y_i = +1 \\ \wedge\, h(x_i) > t}} D_i$$

$$W_-^- = \sum_{\substack{i \,:\, y_i = -1 \\ \wedge\, h(x_i) \le t}} D_i$$

Using this notation,

$$Z = W_+^+ e^{-\alpha} + W_+^- e^{-\beta} + W_-^+ e^{\alpha} + W_-^- e^{\beta} \qquad (8)$$

We can minimize Z with respect to α by taking the partial derivative with respect to α and setting it equal to 0.

$$\frac{\partial Z}{\partial \alpha} = W_-^+ e^{\alpha} - W_+^+ e^{-\alpha} = 0$$

$$\Rightarrow \alpha = \frac{1}{2} log\left(\frac{W_+^+}{W_-^+}\right) \qquad (9)$$

Similarly,

$$\beta = \frac{1}{2} log\left(\frac{W_+^-}{W_-^-}\right). \qquad (10)$$

We use these equations to set α and β on each round of AdaBoost.

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where the labels $y_i \in \{-1, +1\}$ for negative and positive examples respectively. For face recognition, $x_i = (I_1^i, I_2^i)$.

- Initialize weights $D_{1,i} = \frac{1}{n}$ where n is the total number of negative and positive examples.

- Let R be the number of rounds to boost before resampling

- For $t = 1, \ldots, T$:

  1. Normalize the weights,

  $$w_{t,i} \leftarrow \frac{D_{t,i}}{\sum_{j=1}^{n} D_{t,j}}$$

  so that D_t is a probability distribution.

  2. For each filter, ϕ_j, compute the best weak classifier, h_j, that uses ϕ_j. This amounts to finding the optimal threshold t_j minimizing equation 6 for each possible filter. The error, ϵ_j, is defined in equation 6.

  3. Choose the classifier, h_t, with the lowest error ϵ_t.

  4. Choose α and β according to equations 9 and 10. This defines the feature, f_t given in equation 3.

  5. If t is a multiple of R then resample to generate a new training set with new weights.
  Otherwise update the weights:

  $$D_{t+1,i} = D_{t,i} e^{-f_t(x_i)y_i}$$

- The final strong classifier is:

$$F(x) = sign\left(\sum_{t=1}^{T} f_t(x)\right).$$

Figure 2: The AdaBoost algorithm for classifier learning. Each round of boosting selects one feature from the potential features.

3.2 Resampling

Recall that the total number of pairs is $M = \binom{KN}{2}$ which can be prohibitively large. A simple proposal is to take a random subset of the pairs for training. One issue is that the number of positive examples, $K\binom{N}{2}$, is tiny compared to the number of negatives. Chosing random subset with 10% of the pairs, could lead to the inclusion of very few positive examples. Alternatively, selecting more positives than negatives can significantly bias the classifier.

Recall that with each round of boosting the minimum error weak classifier is selected:

$$\min_f \sum_i D_i y_i f(x_i)$$

where $y_i \in \{+1, -1\}$ is the example label and D_i is the normalized example weight. This "expectation" can be approximated by first drawing a sample of examples based on p_i, a proposal distribution. Using this new sample, the expectation is approximated as:

$$\min_f \sum_j \frac{D_j}{p_j} y_j f(x_j)$$

where the subscript j ranges over the sampled subset. If we choose $p_j = D_j$ then the weight on sampled examples is unity. This approach focuses the computation on examples with high weight, while preserving some semblance of the overall distribution.

Resampling can be implemented in two passes as follows [2]. Define the cumulative weight

$$c_k = \sum_{i=0}^{k} D_i$$

where c_M equals the total weight of all examples (where M is the total number of examples). Given a target sample size of S, generate S numbers, r_s on the interval $[0, c_M]$. Example x_j is given weight equal to the number of r_s such that $c_j < r_s < c_{j+1}$. This is equivalent to choosing an example multiple times each with weight 1. Example x_j is chosen for the new training set if its weight is non-zero.

The selection process can be implemented in two passes over the data - once to compute c_M and once to select examples according to the random samples.

4 Results

We have tested our face recognition algorithm on two different test sets. The first is the FERET test set which has been widely used to evaluate face recognition algorithms [8]. The second is a set of face images collected at our lab which contains examples of 36 people over a period of weeks.

4.1 FERET test set

We tested on the FERET set of FA and FB images. The FA images are used as gallery images and the FB images are used as probes. There are 1196 FA images and 1195 FB images. All subjects (with 1 exception) have exactly one gallery and one probe image. The FA and FB images for a single subject vary only in expression (neutral versus smiling).

During the FERET competition one third of the FA/FB data was made available for training, and that training data was included in the final testing set [7]. The identity of the sequestered examples is no longer available, so we use 398

Figure 3: The first 5 features learned by AdaBoost on the FERET training set are shown. The right filter is the negative of the left. A training image pair is also shown for reference.

randomly selected pairs from the FA and FB set for training data. All images are cropped and rectified according to the manually placed eye positions supplied with the FERET data. We scale the images to 45 pixels high by 36 pixels wide.

The training set yields 398 same pairs and 79,003 (398*397/2) different pairs. We use the AdaBoost algorithm with resampling described in section 3. At any one time, only 3178 different pairs and all 398 same pairs are used for training. A new set of 3178 different pairs are chosen from the full training set by resampling after 25 new features have been learned by AdaBoost.

The training algorithm selects from a total 52,374 rectangle filters. These were uniformly sampled from the much larger set of filters that will fit in a 45x36 pixel analysis window.

We ran AdaBoost for a total of 400 rounds yielding a final classifier with 400 features. The first 5 features learned are shown in figure 3. The first filter examines the eye and part of the forehead. The second filter gives a rough indication of the width of the face (how close the edge of the face is the left side of the image). The fifth filter serves a similar purpose on the right side of the image.

Our results compare well with the best reported results on the FERET data [8, 9]. On the recognition problem in which there are 1196 gallery images and 1195 probe images, we achieve a rank-1 recognition rate of 94%. The best reported result on this set is the USC system of von der Malsburg et al. which achieves 96% [8]. The rank-N recognition rate is the percentage of times that the correct gallery image is within the top N similarity scores for each probe image. On the verification problem, we achieve a 1% equal error rate. The equal error rate is the point at which the percentage of correct verifications equals one minus the percentage of false alarms. The best reported equal error rate on the FERET data is also 1% by the University of Maryland system [14, 9], while the USC system has a 2% equal error rate.
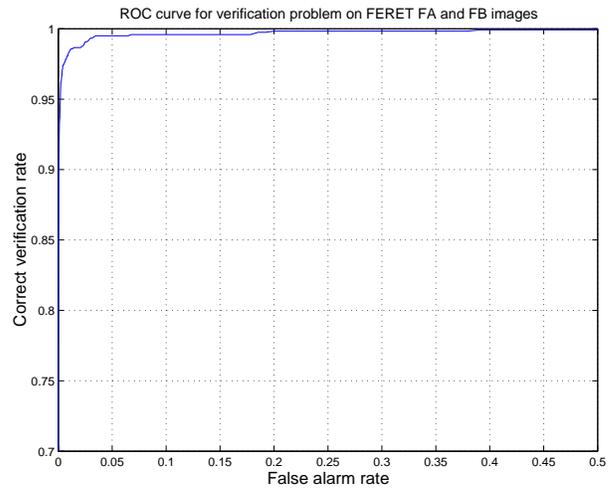

Figure 4: ROC curve for verification task on FERET FA and FB images. The equal error rate is about 1%.

4.2 Lab test set

For security and access control applications, it is important to understand how well a face recognition system performs as faces change subtly (or drastically) from day to day. For such a scenario, we need a large data set of faces taken over many weeks (or longer). We have begun collecting such a data set at our lab. The data set currently consists of 36 people and about 3000 images taken over a period of about a month. A person is photographed under semi-controlled lighting (we have two light stands just behind the camera, but there are windows which yield unpredictable natural lighting as well). In general the lighting is fairly uniform across the face. The data acquisition station is near the entrance of the lab, and individuals can go to this station at any point during the day. In each session 10 images are taken of the same individual using a digital video camera. The person is asked to look at the camera with a neutral expression and then a smile. The data set also contains some images of the same person with and without glasses. Participation is rewarded yet optional, and as a result the quantity of data varies from 1 session to 30 with the average being about 9 sessions. We are continuing to add to this set so it grows daily. A few images for 3 of the subjects are shown in figure 6.

All images were cropped and rectified by automatically detecting the face and both eyes using the face detector of Viola and Jones and left and right eye detectors built using the same technique. The faces were cropped to 45×36 pixels. Each 45 x 36 pixel image is variance normalized to partially compensate for different lighting conditions during image acquisition. Figure 7 illustrates the automatic cropping and rectification stage.
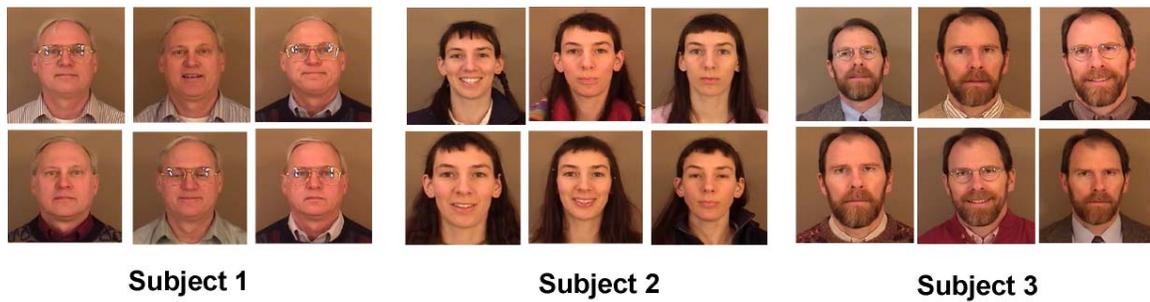
Figure 6: Example faces in our database. Images shown have not been cropped and rectified. Each person is photgraphed over many days.
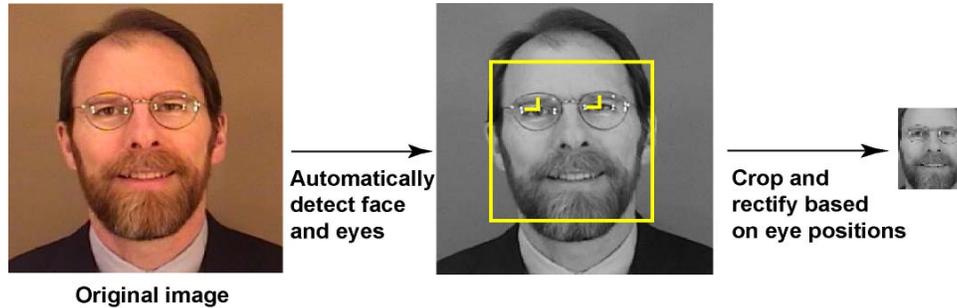

Figure 7: Automatic face and eye detection to crop and rectify faces

The data was divided into a training set consisting of 20 people and a test set consisting of 16 people.

An initial training set of 2500 same face pairs and 2500 different face pairs was sampled from the entire training set. The resampling AdaBoost algorithm was trained for 400 rounds with resampling every 25 rounds. The first 5 filters learned by AdaBoost are shown in figure 8. The first filter is a fairly global measure of the brightness of the top third of the face compared to the middle third. The second filter measures a characteristic of the nose. The third feature seems to be a measure of the hairline and probably indicates that there was not a lot of variation in hair over time within a single subject..

The resulting 400 feature face similarity function was then tested on the 16 person test set consisting of 1239 faces. We again looked at both the face verification problem and the face recognition problem.

For face verification we computed the correct verification rate and the false alarm rate for a large sample (50,000 face pairs) of the total set of same and different face pairs. We achieve a 5.5% equal error rate. The full ROC curve is shown in figure 9.

Although this test set is currently fairly small in terms of the total number of people, it has much more variation than the FERET FA and FB images. Since verification is not dependent on the number of people in the gallery, these results are valid for any sized test set.

For face recognition, we used a single face image in the gallery for each of the 16 subjects. Results can vary significantly depending on which faces are chosen to be in the gallery. To deal with this we chose a random face from each person's set of faces to be in the gallery. Then we computed the rank-N recognition rates using this gallery set and using all other images in the test set as probes. We ran this experiment 100 times and computed the average rank-N recognition rates. The average rank-N rates are shown in figure 10.

5   Conclusions

Face recognition, because of its many applications in automated surveillance and security, has garnered a great deal of attention. While there have been many papers published in this area, much of the debate has now moved outside of the academic arena. Since details of many of the best commercial algorithms are not publically available, it can be difficult to compare results or gauge progress.

One approach for building reliable commercial recognition systems is to combine many disparate recognition al-
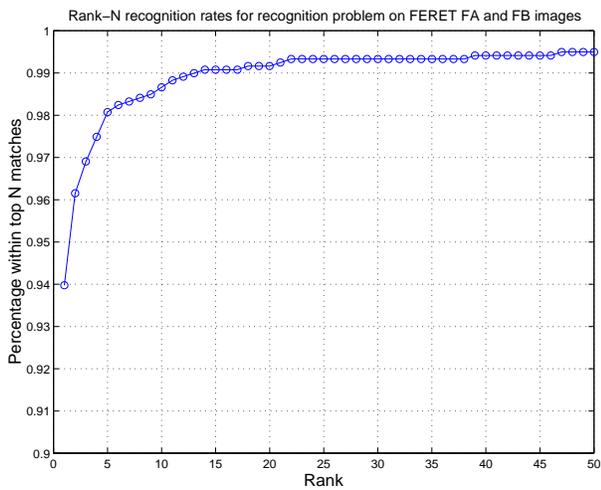
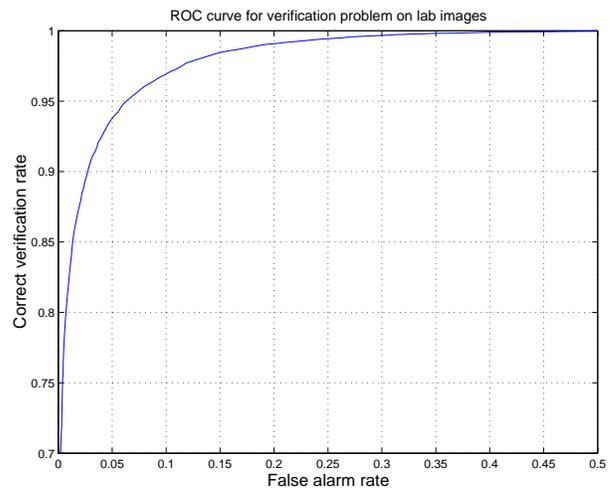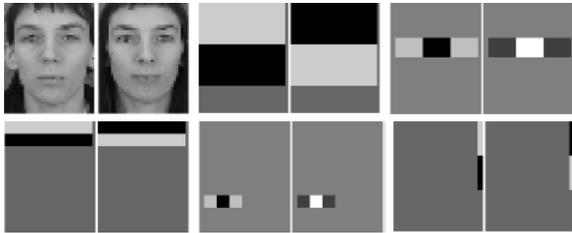Figure 5: Rank-N recognition rates for FERET FA and FB images.

Figure 8: The first 5 features learned by AdaBoost on the lab faces training set are shown. A training image pair is also shown for reference.

gorithms. Since the face recognition problem remains quite difficult, there appears to be a clear need for new and different algorithms. For this reason the appearance of a new algorithm which is computationally efficient should have an immediate impact on practical systems, since it can be included in any system which evalutes a suite of algorithms.

We have presented a novel face recognition system based on simple local features that are learned by looking at intra-personal and extra-personal variations. Our system in comparable to the best reported systems on the FERET database. It also has the advantage of being very computationally efficient and light weight.

The learning algorithm we use to train the face similarity function is also new. It allows AdaBoost to be used on problems that are too large to handle with normal AdaBoost. The resampling idea is not specific to face recognition and so will have broad applicability for any large machine learning problem.

Figure 9: ROC curve for verification task on lab images. The equal error rate is about 5.5%.

References

[1] F. Crow. Summed-area tables for texture mapping. In *Proceedings of SIGGRAPH*, volume 18(3), pages 207–212, 1984.

[2] Yoav Freund. personal communication.

[3] Baback Moghaddam and Alex Pentland. Beyond euclidean eigenspaces: Bayesian matching for visual recognition.

[4] Kazunori Okada, Johannes Steffens, Thomas Maurer, Hai Hong, Egor Elagin, Hartmut Neven, and Christoph von der Malsburg. The Bochum/USC Face Recognition System And How it Fared in the FERET Phase III test. In H. Wechsler, P. J. Phillips, V. Bruce, F. Fogeman Soulié, and T. S. Huang, editors, *Face Recognition: From Theory to Applications*, pages 186–205. Springer-Verlag, 1998.

[5] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, 1998.

[6] P. Penev and J. Atick. Local feature analysis: A general statistical theory for object representation. *Neural Systems*, 7:477–500, 1996.

[7] P. Jonathon Phillips. personal communication.

[8] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Patt. Anal. Mach. Intell.*, 22(10):1090–1104, 2000.

[9] Syed A. Rizvi, P. Jonathon Phillips, and Hyeonjoon Moon. A verification protocol and statistical performance analysis for face recognition algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

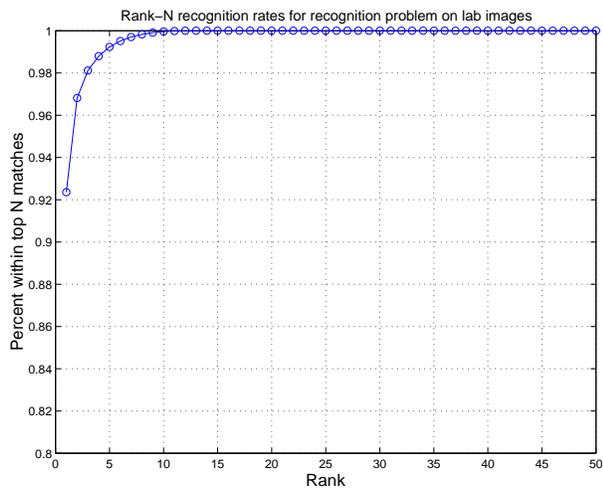[10] R. Schapire and Y. Singer. Improving boosting algorithms using confidence-rated predictions, 1999.

7

Figure 10: Rank-N recognition rates for lab images.

[11] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, HI, December 2001.

[12] P. Viola and M. Jones. Robust real-time object detection. In *Proc. of IEEE workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, July 2001.

[13] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. In Gerald Sommer, Kostas Daniilidis, and Josef Pauli, editors, *Proc. 7th Intern. Conf. on Computer Analysis of Images and Patterns, CAIP'97, Kiel*, number 1296, pages 456–463, Heidelberg, 1997. Springer-Verlag.

[14] W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. In *International Conference on Automatic Face and Gesture Recognition*, pages 336–341, 1998.