IDIAP RESEARCH REPORT

Comparison of MLP and GMM Classifiers for Face Verification on XM2VTS

Fabien Cardinaux * Conrad Sanderson *

Sébastien Marcel *

IDIAP–RR 03-10

January 2003

SUBMITTED FOR PUBLICATION

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

* cardinau,marcel@idiap.ch, conradsand@ieee.org

IDIAP Research Report 03-10

COMPARISON OF MLP AND GMM CLASSIFIERS FOR FACE VERIFICATION ON XM2VTS

Fabien Cardinaux Conrad Sanderson Sébastien Marcel

JANUARY 2003

SUBMITTED FOR PUBLICATION

Abstract. We compare two classifier approaches, namely classifiers based on Multi Layer Perceptrons (MLPs) and Gaussian Mixture Models (GMMs), for use in a face verification system. The comparison is carried out in terms of performance, robustness and practicability. Apart from structural differences, the two approaches use different training criteria; the MLP approach uses a discriminative criterion, while the GMM approach uses a combination of Maximum Likelihood (ML) and Maximum a Posteriori (MAP) criteria. Experiments on the XM2VTS database show that for low resolution faces the MLP approach has slightly lower error rates than the GMM approach; however, the GMM approach easily outperforms the MLP approach for high resolution faces and is significantly more robust to imperfectly located faces. The experiments also show that the computational requirements of the GMM approach can be significantly smaller than the MLP approach at a cost of small loss of performance.

1 Introduction

Identity verification has many real-life applications ranging from access control, transaction authentication (e.g. in telephone banking or remote credit card purchases), to voice mail and secure teleworking.

The goal of an *automatic identity verification system* is to either accept or reject the identity claimed by a given person. Biometric identity verification systems are based on the characteristics of a person, such as their face, fingerprints or signature [16]. Identity verification using face information is an active research area mainly because of its non-intrusive interaction with the users.

The problem of face verification has been addressed by many researchers proposing many different methods. The aim of this paper is not to propose new approaches for face verification, but rather to present a comparison of two popular classification approaches: Multi-Layer Perceptrons (MLPs) and Gaussian Mixture Model (GMMs). In order to obtain comparative results the same feature extraction technique is utilized for both MLP and GMM approaches. The experiments are carried out using faces from the XM2VTS database [9]. To compare the robustness of both approaches, we perform the experiments using two different face image sizes and with manually & automatically located faces.

The rest of this paper is structured as follows. In Section 2 we introduce the reader to the specific problem of face verification. In Section 3 we present a facial feature extraction approach which is suitable for both MLP and GMM based systems. The MLP and GMM classifiers are described in Sections 4 and 5, respectively. Section 6 is devoted to experiments evaluating the two approaches. We analyze the results and draw conclusions in Section 7.

To keep consistency with traditional matrix notation, image sizes are described using the rows first, followed by the columns.

2 Face Verification

An identity verification system has to discriminate between two kinds of events: either the person claiming a given identity is the true claimant (a client) or the person is an impostor.

Generally speaking, a full face verification system can be thought of as being composed of several stages: *image acquisition*, *image processing* (e.g., apply filtering algorithms to reduce the noise), *face detection* and finally *face verification* itself, which usually consists of feature extraction followed by classification.

In many face verification studies it is often assumed that the detection step has been performed perfectly, however, this is not realistic. In this study, results are given for perfect detection as well as in more realistic conditions, i.e., using an automatic face detector.

3 Feature Extraction

In *DCT-mod2* feature extraction [14] a given face image[1] is analyzed on a block by block basis; each block is $N \times N$ (here we use $N = 8$) and overlaps neighboring blocks by 50%. Each block is decomposed in terms of 2D Discrete Cosine Transform (DCT) basis functions [7]. A feature vector for each block is then constructed as:

$$\vec{x} = \left[\ \Delta^h c_0\ \Delta^v c_0\ \Delta^h c_1\ \Delta^v c_1\ \Delta^h c_2\ \Delta^v c_2\ \ c_3\ c_4\ ...\ c_{M-1}\ \right]^T \tag{1}$$

where c_n represents the n-th DCT coefficient, while $\Delta^h c_n$ and $\Delta^v c_n$ represent the horizontal and vertical delta coefficients respectively, and are computed using DCT coefficients extracted from neighboring blocks. Compared to traditional DCT feature extraction [4], the first three DCT coefficients are replaced by their

[1] We use two image sizes: 40×32 and 80×64 (rows $\times$ columns)

respective horizontal and vertical deltas in order to reduce the effects of illumination direction changes. In this study we use M=15 (choice based on [14]), resulting in an 18 dimensional feature vector for each block.

Since *DCT-mod2* feature extraction for a given block is only possible when the block has vertical and horizontal neighbours, processing an image which has Y rows and X columns results in $(2\frac{Y}{N}-3)\times(2\frac{X}{N}-3)$ feature vectors; thus for a 40×32 image, there are $7\times 5 = 35$ vectors.

For the MLP approach, all *DCT-mod2* feature vectors are concatenated to form a composite feature vector, having a dimensionality of $35\times 18 = 630$ for a 40×32 image.

4 MLP Based Classifier

A Multi-Layer Perceptron (MLP) is a particular architecture of Artificial Neural Networks [15]. We will assume that we have access to a training dataset of l pairs $(\vec{x}_i, y_i)$ where $\vec{x}_i$ is a vector containing the pattern, while y_i is the class of the corresponding pattern. For a 2-class task, y_i can be coded as $+1$ and -1.

An MLP is composed of layers of non-linear but differentiable parametric functions. Here we use a MLP with one hidden layer; the hidden and output layers have $\tanh(\cdot)$ transfer functions.

An MLP can be trained by gradient descent using the back-propagation algorithm [15] to optimize any derivable criterion, such as the Mean Squared Error. Here, an MLP is trained (for each client) to classify an input to be either the given client or an impostor. The input of the MLP is a vector corresponding to the features extracted from the face image. The output of the MLP is either 1 (if the input corresponds to a client) or -1 (if the input corresponds to an impostor). The MLP is trained using both client images and impostor images (impostor images are taken to be the images corresponding to other available clients); thus the MLP uses a discriminative training approach.

The decision to accept or reject a client access depends on the score ($\hat{y}$) obtained by the MLP corresponding to the claimed identity; using a threshold t (chosen on a separate validation set), the client is accepted (classified as a true claimant) when $\hat{y}\geq t$, and rejected (classified as an impostor) when $\hat{y} < t$.

5 GMM Based Classifier

Given a claim for person C's identity and a set of feature vectors $X = \{\vec{x}_i\}_{i=1}^{N_V}$ supporting the claim, the average log likelihood of the claimant being the true claimant is calculated using:

$$\mathcal{L}(X|\lambda_C) = \frac{1}{N_V}\sum_{i=1}^{N_V}\log p(\vec{x}_i|\lambda_C) \tag{2}$$

$$\text{where} \quad p(\vec{x}|\lambda) = \sum_{j=1}^{N_G} m_j\,\mathcal{N}(\vec{x};\vec{\mu}_j,\mathbf{\Sigma}_j) \tag{3}$$

$$\lambda = \{m_j,\vec{\mu}_j,\mathbf{\Sigma}_j\}_{j=1}^{N_G} \tag{4}$$

Here, $\mathcal{N}(\vec{x};\vec{\mu},\mathbf{\Sigma})$ is a D-dimensional Gaussian function with mean $\vec{\mu}$ and diagonal covariance matrix $\mathbf{\Sigma}$:

$$\mathcal{N}(\vec{x};\vec{\mu},\mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}}\exp\left[\frac{-1}{2}(\vec{x}-\vec{\mu})^T\mathbf{\Sigma}^{-1}(\vec{x}-\vec{\mu})\right] \tag{5}$$

λ_C is the parameter set for person C, N_G is the number of Gaussians and m_j is the weight for Gaussian j (with constraints $\sum_{j=1}^{N_G} m_j = 1$ and $\forall\, j : m_j \geq 0$).

Given the average log likelihood of the claimant being an impostor, $\mathcal{L}(X|\lambda_{\overline{C}})$, an opinion on the claim is found using:

$$\Lambda(X) = \mathcal{L}(X|\lambda_C) - \mathcal{L}(X|\lambda_{\overline{C}}) \tag{6}$$

The verification decision is reached as follows: given a threshold t, the claim is accepted when $\Lambda(X) \geq t$ and rejected when $\Lambda(X) < t$.

5.1 Model Training

Given a set of training vectors, $X = \{\vec{x}_i\}_{i=1}^{N_V}$ (which may come from several images), the GMM parameters (λ) for each client model are found by adapting a Universal Background Model (UBM) using a form of *maximum a posteriori* (MAP) adaptation [6, 11]. The UBM is trained with the Expectation Maximization (EM) algorithm [2, 3] using training data from all clients.

Since the UBM is a good representation of many clients, it is also used to find the likelihood of the claimant being an impostor, i.e.:

$$\mathcal{L}(X|\lambda_{\overline{C}}) = \mathcal{L}(X|\lambda_{\text{UBM}}) \tag{7}$$

6 Comparison

In this section, we present an experimental comparison between face verification using MLPs and GMMs. This comparison has been done using the multi-modal XM2VTS database and its associated experimental protocol [9]. The MLP and GMM classifiers were implemented using the Torch library [1].

6.1 Database and Protocol

The XM2VTS database contains synchronized video and speech data from 295 subjects, recorded during four sessions taken at one month intervals.

The database is divided into three sets: a training set, an evaluation set and a test set. The training set was used to build client models, while the evaluation set was used to compute the decision thresholds (as well as other hyper-parameters) used in the MLP and GMM approaches. Finally, the test set was used to estimate the performance.

The 295 subjects were divided into a set of 200 clients, 25 evaluation impostors and 70 test impostors. Using Configuration I of the experimental protocol lead to the following setup:

- Training client accesses: 3
- Evaluation client accesses: 600
- Evaluation impostor accesses: $40,000$ ($25 \times 8 \times 200$)
- Test client accesses: 400 (200×2)
- Test impostor accesses: $112,000$ ($70 \times 8 \times 200$)

A verification system can make two types of errors: a *false acceptance* (FA), when the system accepts an *impostor*, and a *false rejection* (FR), when the system rejects a *true client*. The performance of the system is often measured in terms of these two errors, as follows:

$$\text{FAR} = \frac{\text{number of FAs}}{\text{number of impostor accesses}} \times 100\% \tag{8}$$

$$\text{FRR} = \frac{\text{number of FRs}}{\text{number of client accesses}} \times 100\% \tag{9}$$

Since in real life the decision threshold has to be chosen *a priori*, it is selected to obtain Equal Error Rate (EER) performance (where FAR=FRR) on the validation set; it is then used on the test set to obtain the final performance figure. In order to combine FAR and FRR into one number, Half Total Error Rate (HTER) can be used:

$$\text{HTER} = \frac{\text{FAR} + \text{FRR}}{2} \tag{10}$$

6.2 Experiment Setup

For each client model, the training set is composed of a client training set (3 images) and an impostor training set. Each image was normalized in size via an affine transformation [7]; the normalized interocular distance was selected so that a face window with a resolution of 80×64 (or 40×32) contained the face area from the eyebrows to the mouth; moreover, the location of the eyes was the same in each face window.

Since there is not enough face images in the training set to adequately train each MLP, the training set was artificially extended (as done in comparable studies [8]). For fair comparison purposes, the same extended training set was also used to train the client models in the GMM approach (theoretically, the GMM approach does not require the extended training set - see also Section 6.4).

The client training set was enlarged by shifting (8 directions and 4 pixel shifts), scaling (2 scales) and mirroring each original face window. Hence the training set for each person contains $3P = 990$ face windows, where $P = 2AB$, i.e. the number of shifted & scaled face patterns and their mirrored versions. Here, $A = \text{number of shifts} \times 8 + 1$ is the total number of shifts, in 8 directions, including the original face window; $B = \text{number of scales} \times 2 + 1$ is the total number of scales, in 2 directions (down-scaling and up-scaling), including the original size. Each MLP was also trained using a set of pseudo-impostors composed of the other 199 clients and their mirrored images; thus the pseudo-impostor training set contained $199 \times 3 \times 2 = 1194$ patterns.

For the GMM approach, the UBM was trained using all client training set and their mirrored images; Client models were derived from the UBM using the extended client training set.

In order to find the optimal capacity of the models, we used the evaluation set to select the size of the model (number of Gaussians for the GMM approach and number of hidden units for the MLP approach) as well as other hyper-parameters such as the variance floor for the UBM and the learning rate for the MLP approach.

For each value of the hyper-parameter to tune, we trained the client models using the extended training set. We then selected the value of the hyper-parameter that optimized the EER on the evaluation set. Finally, we tested the models using these hyper-parameters on the test set.

6.3 Experiments

In the first experiment we compared the performance of the two approaches with face windows extracted using manually located eye coordinates; the face windows had a resolution of 80×64. The values of the hyper-parameters are as follows: 512 Gaussians for GMMs approach and 128 hidden units in MLP approach. The Detection Error Tradeoff (DET) curves [10] of the two approaches are shown in Figure 1.

The second experiment was similar to the first; here the face windows had a resolution of 40×32. The values of the hyper-parameters are as follows: 512 Gaussians for the GMM approach and 32 hidden units in MLP approach. The DET curves are shown in Figure 2.

Experiments 3 and 4 were similar to 1 and 2, respectively; here, the eye locations were found using a simple face detector [13]. The same values of the hyper-parameters were used as found for face windows extracted using manually located eye coordinates. The DET curves are shown in Figure 3 for 80×64 faces and in Figure 4 for 40×32 faces.

The corresponding FAR, FRR and HTER for all experiments are given in Tables 1 and 2. The computation times to train and test all 200 clients are given in Table 3.

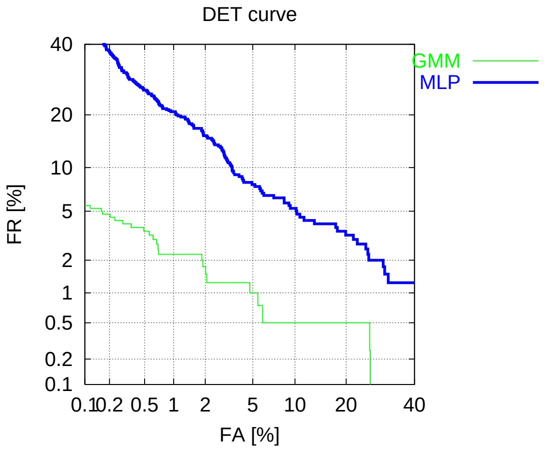

Figure 1: DET curves for *manually* located 80×64 faces

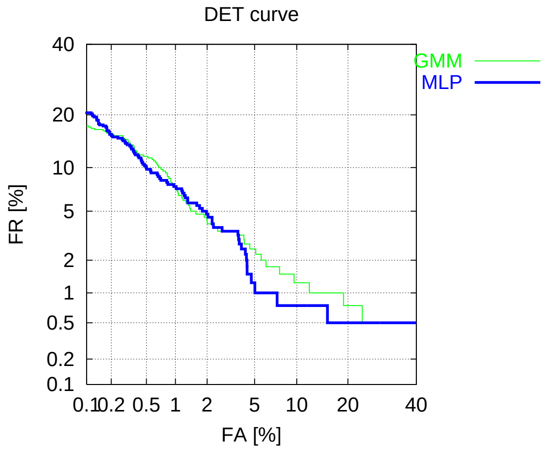

Figure 2: DET curves for *manually* located 40×32 faces

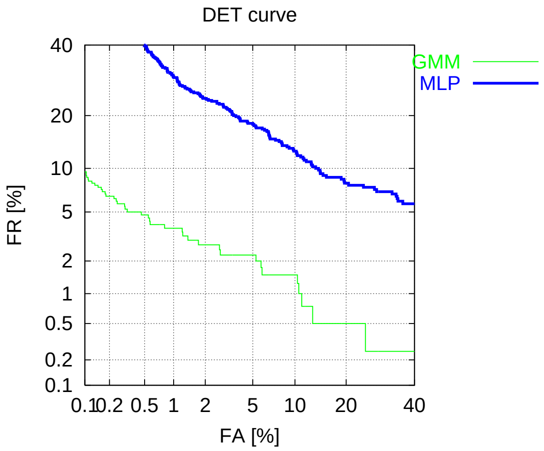

Figure 3: DET curves for *automatically* located 80×64 faces

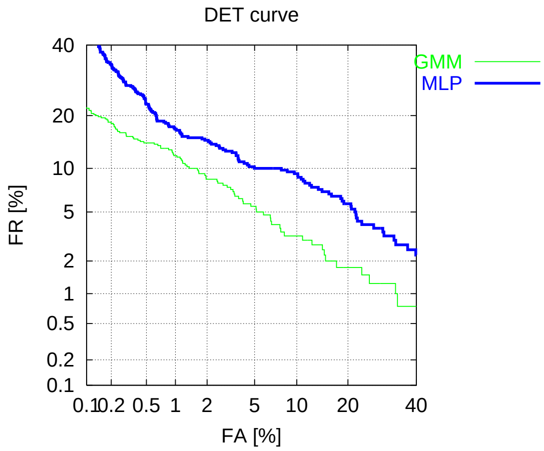

Figure 4: DET curves for *automatically* located 40×32 faces

Model type (face size)	*FAR*	*FRR*	*HTER*
GMM (80×64)	1.69	2.25	1.97
MLP (80×64)	4.44	8.00	6.22
GMM (40×32)	4.84	2.50	3.67
MLP (40×32)	3.22	3.50	3.36

Table 1. Performance of MLP and GMM approaches using *manually* located faces

Model type (face size)	*FAR*	*FRR*	*HTER*
GMM (80×64)	2.15	2.75	2.45
MLP (80×64)	11.55	11.25	11.40
GMM (40×32)	5.92	4.75	5.33
MLP (40×32)	7.98	9.75	8.86

Table 2. Performance of MLP and GMM approaches using *automatically* located faces

Model type (face size)	*Time*
GMM (80×64)	2710
MLP (80×64)	2252
GMM (40×32)	386
MLP (40×32)	162

Table 3. Time taken (minutes) to train (using extended training set) and test 200 clients (Pentium IV, 1.6 GHz)

Data (face size)	*FAR*	*FRR*	*HTER*	*Time*
Ext. (80×64)	1.69	2.25	1.97	2710
Orig. (80×64)	2.80	2.25	2.53	80
Ext. (40×32)	4.84	2.50	3.67	386
Orig. (40×32)	8.25	5.75	7.00	3

Table 4. Performance of the GMM approach using extended and original training sets (manually located faces)

6.4 Experiments Using Only the Original Training Set

In order to provide a fair comparison with the MLP approach, in Section 6.3 we used an artificially extended training set for the GMM approach. Since training by MAP adaptation (in the GMM approach) is less sensitive to small amounts of training data than training by back-propagation (in the MLP approach), we have evaluated the performance of the GMM approach when the client models were trained using only the original training set.

The DET curves comparing performance for the extended and the original training sets are shown in Figures 5 and 6 for 80×64 and 40×32 faces, respectively (the eye positions were located manually). The corresponding FAR, FRR, HTER and computation times are given in Table 4. The values of the hyper-parameters are as follows: 512 Gaussians for 80×64 face windows and 64 Gaussians for 40×32 face windows.

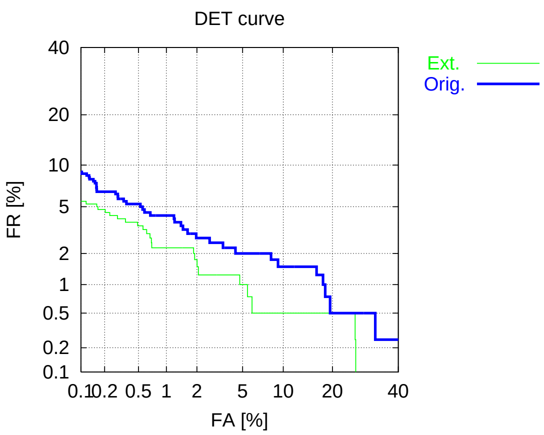

Figure 5: Performance of GMM approach for 80×64 faces (manually located), using original and extended training sets

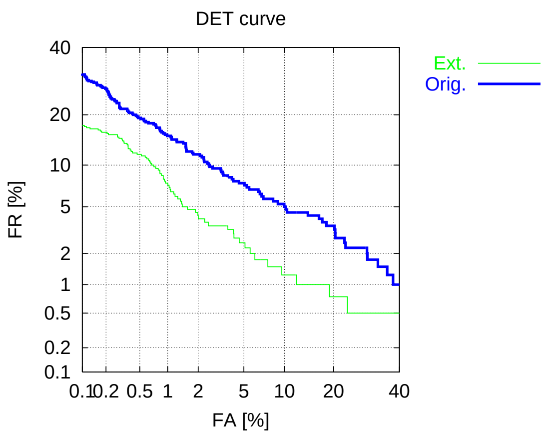

Figure 6: Performance of GMM approach for 40×32 faces (manually located), using original and extended training sets

7 Discussion and Conclusions

Assuming perfect face localisation (where the eye coordinates are manually located), the results show (Figures 1 & 2) that the GMM approach (using face windows with a resolution of 80×64) obtains the lowest error rates. Both MLP and GMM approaches obtain worse results for 40×32 face windows, with the MLP approach obtaining slightly lower error rates than the GMM approach (which can be explained by discriminant training of MLPs). For the 80×64 face windows, the GMM approach easily outperforms the MLP approach; one possible explanation of this result is that for this face size, the generalization performance of the MLP approach is limited by the number of available training patterns. Due to the large dimensionality of the feature vectors used in the MLP approach, a much larger number of training patterns is required to adequately train each MLP.

Using automatic face localisation (Figures 3 & 4), the GMM approach outperforms the MLP approach for both image sizes. Moreover, when comparing the error rates for manually and automatically located faces (Tables 1 & 2), the HTER for the GMM approach increases from 1.97 to only 2.45 for the larger image size and from 3.67 to only 5.33 for the smaller image size, whereas the HTER for the MLP approach increases from 6.22 to 11.40 for the larger image size and from 3.36 to 8.86 for the smaller image size. These results thus suggest that the GMM approach is significantly more robust than the MLP approach to imperfect face localisation.

The above difference can be explained as follows. For the MLP approach the *DCT-mod2* feature vectors extracted from a face window are concatenated to form one large feature vector, thus preserving the location of face characteristics (e.g. eyes and nose); large translations of the face window significantly alter the location of the characteristics in the large feature vector, causing a mismatch between training and test data, which in turn leads to worse performance. In the GMM approach, the location of face characteristics is lost, thus translations of the face window have only minor contribution to the average log likelihood [see Eqn. (2)].

From a real-life application point of view, we note that in the MLP approach each MLP is trained to discriminate between one client and all others clients (pseudo-impostors); if a new client is to be added to the database, the MLP approach requires the pseudo-impostor training data to be present. This is in contrast to the GMM approach, where a new client model is created simply by adapting the UBM, using only the client training data; the UBM can be supplied in a pre-trained form (for example, on a very large data set). We also note that the performance and robustness advantages of the GMM approach come at a similar computational cost to the MLP approach (see Table 3).

Using only the original training set for training the GMM client models results in a small performance decrease for both window sizes (Table 4); however, for 80×64 faces the performance is still better than the MLP approach. Moreover, the computational cost is significantly reduced: it is over 30 times less for 80×64 faces and over 100 times less for 40×32 faces.

8 Acknowledgements

The authors 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)”. This work was also funded by the European projects “BANCA and CIMWOS”, through the Swiss Federal Office for Education and Science (OFES).

References

[1] Collobert, R., Bengio, S., and Mariéthoz, J.: Torch: a modular machine learning software library. IDIAP Research Report **02-46** (2002), Martigny, Switzerland. (see also www.torch.ch)

[2] Dempster, A.P., Laird, N.M., and Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Royal Statistical Soc., Ser. B **39** (1977) 1–38.

[3] Duda, R. O., Hart, P. E., and Stork, D. G.: Pattern Classification. John Wiley & Sons, USA, 2001.

[4] Eickeler, S., Müller, S., Rigoll, G.: Recognition of JPEG Compressed Face Images Based on Statistical Methods. Image and Vision Computing **18** (2000) 279–287.

[5] Féraud, R., Bernier, O., Viallet, J.-E., and Collobert, M.: A Fast and Accurate Face Detector Based on Neural Networks. Trans. Pattern Analysis and Machine Intell. **23** (2001) 42–53.

[6] Gauvain, J-L., and Lee, C-H.: Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. IEEE Trans. Speech and Audio Processing **2** (1994) 291–298.

[7] Gonzales, R. C., and Woods, R. E.: Digital Image Processing. Addison-Wesley, Reading, Massachusetts, 1993.

[8] Kittler, J., Matas, G., Jonsson, K., and Sanchez, M. U. R.: Combining Evidence in Personal Identity Verification Systems. Pattern Recognition Letters **18** (1997) 845–852.

[9] Lüttin, J., and Maître, G.: Evaluation Protocol for the Extended M2VTS Database (XM2VTSDB). IDIAP Communication **98-05** (1998), Martigny, Switzerland.

[10] Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M.: The DET Curve in Assessment of Detection Task Performance. Proc. Eurospeech'97, 1997, pp. 1895–1898.

[11] Reynolds, D., Quatieri, T., and Dunn, R.: Speaker Verification Using Adapted Gaussian Mixture Models. Digital Signal Processing **10** (2000) 19-41.

[12] Rowley, H. A., Baluja, S., and Kanade, T.: Neural Network-Based Face Detection. Trans. Pattern Analysis and Machine Intelligence **20** (1998) 23–38.

[13] Sanderson, C.: Automatic Person Verification Using Speech and Face Information. PhD Thesis, Griffith University, Brisbane, Australia, 2002.

[14] Sanderson, C., and Paliwal, K.K.: Polynomial Features for Robust Face Authentication. Proc. International Conf. on Image Processing, Rochester, New York, 2002, pp. 997-1000 (Vol. 3).

[15] Schalkoff, R. J.: Pattern recognition: statistical, structural and neural approaches. John Wiley & Sons, USA, 1992.

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