Face Verification Using Adapted Generative Models

Fabien Cardinaux, Conrad Sanderson, Samy Bengio
IDIAP, Rue du Simplon 4, CH-1920 Martigny, Switzerland
{cardinau, bengio}@idiap.ch, conradsand@ieee.org

Abstract

It has been shown previously that systems based on local features and relatively complex generative models, namely 1D Hidden Markov Models (HMMs) and pseudo-2D HMMs, are suitable for face recognition (here we mean both identification and verification). Recently a simpler generative model, namely the Gaussian Mixture Model (GMM), was also shown to perform well. In this paper we first propose to increase the performance of the GMM approach (without sacrificing its simplicity) through the use of local features with embedded positional information; we show that the performance obtained is comparable to 1D HMMs. Secondly, we evaluate different training techniques for both GMM and HMM based systems. We show that the traditionally used Maximum Likelihood (ML) training approach has problems estimating robust model parameters when there is only a few training images available; we propose to tackle this problem through the use of Maximum a Posteriori (MAP) training, where the lack of data problem can be effectively circumvented; we show that models estimated with MAP are significantly more robust and are able to generalize to adverse conditions present in the BANCA database.

1. Introduction

Identity verification using face images is an active research area and has many real-life applications, such as access control, transaction authentication and secure teleworking. An identity verification system has to discriminate between two kinds of events: either the person claiming a given identity is the true claimant or the person is an impostor. This is in contrast to an identification system, which attempts to find the identity of a given person out of a pool of people. Both verification and identification systems can be thought of as falling in the general research area of face recognition.

Many techniques have been proposed for face recognition; some examples are systems based on PCA-based feature extraction (eigenfaces), Elastic Graph Matching, Artificial Neural Networks [17], Support Vector Machines and Normalised Correlation [13]. Examples specific to generative models include 1D Hidden Markov Models (HMMs) [14], pseudo-2D HMMs [6, 10] and Gaussian Mixture Models (GMMs) [2, 15, 16] (which can be considered as a simplified version of HMMs). All of the above-mentioned generative models use local features (that is, the features only describe a part of the face); this is in contrast to holistic features, such as in the PCA-based approach, where one feature vector describes the entire face.

In generative approaches, the face is typically analyzed on a block by block basis, and feature extraction such as 2D DCT [8] or DCTmod2 [15] is applied to each block. In the HMM approaches, the spatial relation between major face features (such as the eyes and nose) is kept (although not rigidly); in the GMM approach the spatial relation is effectively lost (as each block is treated independently), resulting in good robustness to imperfectly located faces [2]. In this paper we first propose to restore some of spatial relation by using local features with embedded positional information. By working in the feature domain, the simplicity advantage of the GMM approach is retained.

In the approaches presented in [6, 10, 14, 15], generative models are trained using the Maximum Likelihood (ML) criterion via the Expectation Maximization (EM) algorithm [4]. It is generally known that one of the drawbacks of training via this paradigm is that a lot of data is required to properly estimate model parameters; this can be a problem when there are only a few training images available. In an attempt to tackle this problem, Eickeler *et al.* [6] proposed to use a well trained generic (non-client specific) model as the starting point for ML training. While the results in [6] were promising, they were obtained on the rather easy ORL database [14]. Through experiments on the much harder BANCA database [1], we will show that even with the generic model as the starting point, ML training still produces poor models. Our second main proposition is thus to replace ML training with Maximum *a Posteriori* (MAP) training [7], which effectively circumvents the lack of data problem.

The tone of this paper is hence an evaluation, on a common database, of different approaches to face verification using generative models. In Section 2 we briefly overview the BANCA database and its experiment protocols. In Section 3 we summarize the DCTmod2 feature extraction and describe the proposed extension (embedding of positional information). In Section 4 we review the

GMM, 1D HMM and pseudo-2D HMM representations of faces; we also describe MAP training for each model. Section 5 is devoted to experiments; here we evaluate the GMM approach using standard DCTmod2 and the proposed extended features; we also evaluate the GMM, 1D and pseudo-2D HMM approaches trained with traditional ML, ML initialized by a global model, and the suggested MAP approach. We analyze the results, draw conclusions and suggest future work in Section 6.

2. The BANCA Database and Protocols

The BANCA database [1] was designed to test multi-modal identity verification with various acquisition devices under several scenarios. In our experiments we use face images from the English corpus which contains 52 subjects; the population is subdivided into 2 groups of 26 subjects, denoted as $g1$ and $g2$.

Each subject participated in 12 recording sessions in different conditions and with different cameras. Each of these sessions contains two video recordings: one true client access and one impostor attack. Five "frontal" (not necessarily directly frontal) face images have been extracted from each video recording. Sessions 1-4 contain data for the *controlled* condition, while sessions 5-8 and 9-12 respectively contain *degraded* and *adverse* conditions (see Fig. 1); the latter two conditions differ from the *controlled* condition in terms of image quality, lighting and/or pose.

Seven distinct configurations specify which images can be used for training and testing; the configurations are: Matched Controlled (Mc), Matched Degraded (Md), Matched Adverse (Ma), Unmatched Degraded (Ud), Unmatched Adverse (Ua), Pooled test (P) and Grand test (G). Table 1 describes the usage of different sessions in each configuration.

We believe that the most realistic cases are when we train the system in controlled conditions and test it in different conditions; hence in this paper we only performed experiments with configurations Mc, Ud, Ua and P. This limitation to four different scenarios should make the results easier to interpret. Performance is measured in terms of Half Total Error Rate (HTER), defined as:

$$\text{HTER} = (\text{FAR} + \text{FRR})\,/2$$

where FAR and FRR are the False Acceptance Rate and False Rejection Rate, respectively. 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 a HTER figure. Here we use set $g1$ as the validation set and set $g2$ as the test set.

Figure 1. Example images from the BANCA database. Left to right: *controlled*, *degraded* and *adverse* conditions.

3. Preprocessing and Feature Extraction

Face images from the BANCA database are converted into gray-scale values and a 80×64 (rows $\times$ columns) face window is cropped out; face location is based on manually located eye positions. Each face window contains the face area from the eyebrows to the mouth; moreover, the location of the eyes was the same in each face window (via geometric normalization); see Fig. 2 for an example.

As mentioned before we use manually located eye positions in order to make the results independent of the quality of the face localization system; however it must be noted that the results are biased when compared to a real life system (where the face needs to be automatically located).

Histogram equalization is used to normalize the face images photometrically. We then extract *DCTmod2* features from each image face [15]. We have found the combination of histogram equalization and feature extraction to provide good results in preliminary experiments. The feature extraction process is summarized as follows. A given face image is analyzed on a block by block basis; each block is $N \times N$ (here we use $N = 8$) and overlaps neighboring blocks by a configurable amount of pixels. Each block is decomposed in terms of two-dimensional DCT basis functions [8]. A feature vector for a block located at row a and column b is then constructed as:

$$\vec{x}_{(a,b)} = \left[\ \Delta^h c_0\ \Delta^v c_0\ \Delta^h c_1\ \Delta^v c_1\ \Delta^h c_2\ \Delta^v c_2\ \ c_3\ c_4\ \ldots\ c_{M-1}\ \right]^T$$

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 [6, 10], the first three DCT coefficients are replaced by their 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 [15]), resulting in an 18 dimensional feature vector for each block.

Test Sessions	Train Sessions			
	1	5	9	1,5,9
C: 2-4 I: 1-4	Mc			
C: 6-8 I: 5-8	Ud	Md		
C: 10-12 I: 9-12	Ua		Ma	
C: 2-4,6-8,10-12 I: 1-12	P			G

Table 1. Usage of the seven BANCA protocols (C: client, I: impostor)

The degree of overlap has two effects: the first is that as overlap is increased the spatial area used to derive one feature vector is decreased; the second is that as the overlap is increased the number of feature vectors extracted from an image grows in a quadratic manner.

3.1. Embedding Positional Information

The above DCTmod2 feature extraction has been successfully used in a GMM based face verification system [2, 15]. In such a GMM system (see Section 4.1 for more details) the spatial relation between major face features (such as the eyes and nose) is effectively lost (as each block is treated independently). We propose to increase the performance of the GMM approach (without sacrificing its simplicity) through extending the DCTmod2 approach with embedded positional information. Formally, the feature vector for a block at row a and column b is found with:

$$\vec{x}_{(a,b)}^{\,\texttt{extended}} = \left[\begin{array}{ccc} \left(\vec{x}_{(a,b)}^{\,\texttt{original}} \right)^T & a & b \end{array} \right]^T$$

where $\vec{x}_{(a,b)}^{\,original}$ is the standard DCTmod2 feature vector for the block located at row a and column b. By explicitly embedding positional information into each feature vector, a weak constraint is placed on the areas that each gaussian in the GMM can model, thus making a face model more specific.

4. Generative Model Based Classifiers

Let us denote the parameter set for client C as λ_C, and the parameter set describing a generic face (non-client specific) as $\lambda_{\overline{C}}$. Given a claim for client C's identity and a set of feature vectors $X = \{\vec{x}_t\}_{t=1}^{N_V}$ supporting the claim (extracted from the given face), we find an opinion on the claim using:

$$\Lambda(X) = \log P(X|\lambda_C) - \log P(X|\lambda_{\overline{C}}) \tag{1}$$

where $P(X|\lambda_C)$ is the likelihood of the claim coming from the true claimant and $P(X|\lambda_{\overline{C}})$ is the likelihood of the claim coming from an impostor. The generic face model is also known as a *universal background model* and as a *world model*; it is typically trained with data from many people (here we use data from BANCA's *Spanish corpus*). The verification decision is then reached as follows: given a threshold τ, the claim is accepted when $\Lambda(X) \geq \tau$ and rejected when $\Lambda(X) < \tau$.

We use three different ways to train each client model:

1. Traditional ML training, where k-means initialization is used [4, 5].
2. ML training with a generic (non-client specific) model as the starting point (as in [6]); data from many people is used the find the parameters of the generic model via traditional ML training; this is the same generic model used for calculating $P(X|\lambda_{\overline{C}})$ in Eqn. (1) for all generative approaches.
3. MAP training [7]; here a generic model is used as in point (2) above, but instead of using it merely as a starting point, the model is *adapted* using client data. Given a set of training vectors X, the probability density function (pdf) $P(X|\lambda)$ and the prior pdf of λ, $P(\lambda)$, the MAP estimate of model parameters, $\lambda_{\texttt{MAP}}$ is defined as:

$$\begin{aligned} \lambda_{\texttt{MAP}} &= \arg\max_{\lambda} P(\lambda|X) & (2) \\ &= \arg\max_{\lambda} P(X|\lambda)P(\lambda) & (3) \end{aligned}$$

Assuming λ to be fixed but unknown is equivalent to having a non-informative $P(\lambda)$, reducing the solution of $\lambda_{\texttt{MAP}}$ to the standard ML solution. Thus, the difference between ML and MAP training is in the definition of a prior distribution for the model parameters to be estimated. Further discussion on MAP training is given in Section 4.1.

4.1. Gaussian Mixture Model

In the GMM approach, the likelihood of a set of feature vectors is found with

$$P(X|\lambda) = \prod_{t=1}^{N_V} P(\vec{x}_t|\lambda) \tag{4}$$

where

$$\begin{aligned} P(\vec{x}|\lambda) &= \sum_{j=1}^{N_G} m_j \, \mathcal{N}(\vec{x}|\vec{\mu}_j, \mathbf{\Sigma}_j) & (5) \\ \lambda &= \{m_j, \vec{\mu}_j, \mathbf{\Sigma}_j\}_{j=1}^{N_G} & (6) \end{aligned}$$

Here, $\mathcal{N}(\vec{x}|\vec{\mu}, \mathbf{\Sigma})$ is a D-dimensional gaussian function with mean $\vec{\mu}$ and diagonal covariance matrix $\mathbf{\Sigma}$. λ is the given parameter set, 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 $\forall\, j : m_j \geq 0$).

An implementation of MAP for client model adaptation consists of using a global parameter to tune the relative importance of the prior. In this case, the equation for adaptation of the means is [9]:

$$\hat{\mu}_k = (1-\alpha)\mu_k^w + \alpha \frac{\sum_{t=1}^{T} P(k|\vec{x}_t)\vec{x}_t}{\sum_{t=1}^{T} P(k|\vec{x}_t)} \tag{7}$$

where $\hat{\mu}_k$ is the new mean of the k-th gaussian, μ_k^w is the corresponding mean in the generic model, $P(k|\vec{x}_t)$ is the posterior probability of the k-th gaussian (from the client model from the previous iteration), and $\alpha \in [0,1]$ is the adaptation factor chosen empirically on a separate validation set. The adaptation procedure is iterative, thus an initial client model is required; this is accomplished by copying the generic model.

As can be seen, the new mean is simply a weighted sum of the prior mean and new statistics; α can hence be interpreted as the amount of faith we have in the new statistics; when the amount of training data is low, we would generally set α to be low.

It must be noted that only the means of the gaussians are adapted, as it has been empirically observed that adaptation of the other parameters generally does not improve performance [9]. The other parameters (the weights and covariance matrices) are copied from the generic model to each client model.

4.2. 1D Hidden Markov Model

The one-dimensional HMM (1D HMM) is a particular HMM topology where only self transitions or transitions to the next state are allowed. This type of HMM is also known as a top-bottom HMM [14] or left-right HMM in the context of speech recognition [12]. Here the face is represented as a sequence of overlapping *rectangular* blocks from top to bottom of the face (see Fig. 2 for an example). To simulate the *rectangular* block representation, DCTmod2 feature vectors from the same line of blocks are concatenated to form a large observation vector.

The model is characterized by the following:

1. N, the number of states in the model; each state corresponds to a region of the face; $S = \{S_1, S_2, \ldots, S_N\}$ is the set of states. The state of the model at row t is given by $q_t \in S$, $1 \leq t \leq T$, where T is the length of the observation sequence (number of rectangular blocks).
2. The state transition matrix $A = \{a_{ij}\}$. The topology of the 1D HMM allows only self transitions or transitions to the next state:

$$a_{ij} = \begin{cases} P(q_t = S_j | q_{t-1} = S_i) & \text{for } j = i,\ j = i+1 \\ 0 & \text{otherwise} \end{cases}$$

3. The state probability distribution $B = \{b_j(\vec{x}_t)\}$, where

$$b_j(\vec{x}_t) = p(\vec{x}_t | q_t = S_j) \tag{8}$$

The features are expected to follow a continuous distribution and are modeled with mixtures of gaussians.

In compact notation, the parameter set of the 1D HMM is:

$$\lambda = (A, B) \tag{9}$$

If we let Q to be a state sequence $q_1, q_2 \cdots q_T$, then the likelihood of an observation sequence X is:

$$P(X|\lambda) = \sum_{\forall\, Q} P(X, Q|\lambda) \tag{10}$$

$$= \sum_{\forall\, Q} \prod_{t=1}^{T} b_{q_t}(\vec{x}_t) \prod_{t=2}^{T} a_{q_{t-1}, q_t} \tag{11}$$

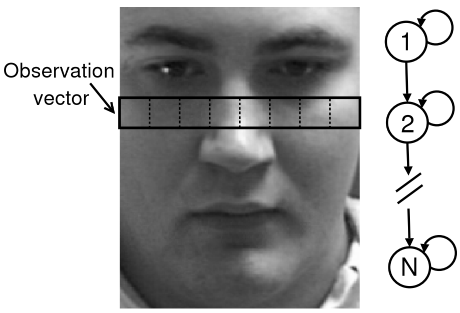

Figure 2. **Sampling window and 1D HMM topology.**

The calculation of this likelihood according to the direct definition (11) involves an exponential number of computations; in practice the Forward-Backward procedure is used [12]; it is mathematically equivalent, but significantly more efficient.

For the case of the 1D-HMM, MAP adaptation of the means is [c.f. Eqn. (7)]:

$$\hat{\mu}_{k,i} = (1-\alpha)\mu^{w}_{k,i} + \alpha \frac{\sum_{t=1}^{T} P(k, i|\vec{x}_t)\vec{x}_t}{\sum_{t=1}^{T} P(k, i|\vec{x}_t)} \tag{12}$$

where $P(k, i|\vec{x}_t)$ is the joint posterior of the state i and its k-th gaussian.

4.3. Pseudo-2D HMM

Emission probabilities of 1D HMMs are typically represented using mixtures of gaussians. For the case of pseudo 2D HMM (P2D HMM) (also known as Embedded HMM [11]) the emission probabilities of the HMM (now referred to as "main HMM") are estimated through a secondary HMM (referred to as an "embedded HMM"). The states of the embedded HMMs are modeled by a mixture of gaussians. This approach was used for the face identification task in [6, 14] and the training process is described in detail in [11]. As shown in Fig. 3, we chose to perform the vertical segmentation of the face image by the main HMM and horizontal segmentation by embedded HMMs. We made this choice because the main decomposition of the face is instinctively from top to bottom (forehead, eyes, nose, mouth).

The corresponding equation for MAP adaptation of the means [c.f. Eqns. (7) and (12)] is:

$$\hat{\mu}_{k,i,j} = (1-\alpha)\mu^{w}_{k,i,j} + \alpha \frac{\sum_{t=1}^{T} P(k, i, j|\vec{x}_t)\vec{x}_t}{\sum_{t=1}^{T} P(k, i, j|\vec{x}_t)} \tag{13}$$

where $P(k, i, j|\vec{x}_t)$ is the joint posterior of the state i of the main HMM, state j of its embedded HMM and its k-th gaussian.

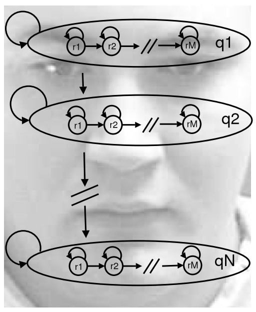

Figure 3. **P2D HMM: the emission distributions of the vertical HMM are estimated by horizontal HMMs. q_i represent the states of the main HMM and r_j represent the embedded HMMs states.**

5. Experiments

For each client model, the training set was composed of five images per client; we artificially increased this to ten images by mirroring each original face window.

The generic model was trained with faces from the Spanish corpus of BANCA (containing faces different from the English corpus) making the generic model independent of the subjects present in the client database.

DCTmod2 features were extracted using either four or seven pixel overlap; using the validation set $g1$ we found that an overlap of four pixels is better for the GMM approaches while an overlap of seven pixels is better for the HMM based approaches. The effects of the differences in the overlap are currently under further investigation.

As described in Section 4.2, feature vectors from the same row are concatenated in the 1D HMM approach. Since the resulting vector would be too big if we concatenate all the features from the same row (recall that seven pixel overlap is used), we chose to concatenate features from every eighth block (thus eliminating horizontally overlapped blocks).

In order to find the optimal capacity of the models, we used the validation set $g1$ to select the size of the model (e.g. number of gaussians in the GMMs and number of states of the HMMs) as well as other hyper-parameters such as the variance floor for the generic model and the adaptation coefficient α. For each value of the hyper-parameter to tune, we trained the client models using the client training set (extended by mirroring); we then selected the value of the hyper-parameter that optimized the EER on the validation set $g1$. Finally, we tested the models using these hyper-parameters on the test set $g2$.

5.1. Results and Discussion

Table 2 shows the results in terms of HTER for the four different systems presented in this paper. Specifically, GMM indicates the GMM approach with standard DCTmod2 feature vectors, GMMext indicates the GMM approach with extended DCTmod2 feature vectors, and 1D HMM & P2D HMM are self explanatory. For all four systems results are shown for the three different training strategies (Section 4); models trained using the traditional ML criterion have a *ML* suffix; for ML training initialized with a generic model, the suffix is *init*; for MAP training, the suffix is *adapt*. The results table also contains performance figures for the best two systems reported in [13]; the first system is a combination of Linear Discriminant Analysis and Normalised Correlation (LDA/NC), while the second is based on Support Vector Machines (SVMs). It must be noted that in [13], $g1$ and $g2$ were used alternatively as the validation set and the test set; the results were then computed using the mean of HTERs from the two configurations; in contrast we have performed our experiments only with the $g1$ as the validation set and $g2$ as the test set.

It is interesting to see that for the Matched Controlled condition (Mc), ML training performs better than adaptation (except for P2D HMM as discussed later) but for Unmatched conditions (Ud and Ua) or partially unmatched condition (P) the models trained by MAP always perform better. We believe that the models trained by ML are too highly tuned (i.e. over-fitted) to the training data (and hence the training condition); while this works well when the training and testing conditions are matched (as in Mc), when the condition changes there is a mismatch between the model and the given condition, resulting in rapid performance degradation. The results also show that ML training with initialization by a generic model generally does not eventuate in better models compared to traditional ML training (where k-means initialization is used).

For the GMM approach, we can see that the use of extended DCTmod2 feature vectors results in better performance compared to standard DCTmod2, especially in the Ua condition. It can also be seen that the performance of the extended GMM approach is comparable to the 1D HMM approach.

In the 1D HMM approach, the dimensionality of the feature vectors is 144 against 18 for the standard GMM and P2D HMM approaches and 20 for the extended GMM approach. We believe that the large dimensionality of the feature vectors used in the 1D HMM approach is the main drawback; the larger the dimensionality, the more training data is required to properly estimate model parameters [5] (especially for the generic model, which is then adapted for each client).

For ML training, the performance of P2D HMM

Protocol	Mc	Ud	Ua	P
LDA/NC (from [13])	4.9	16.0	20.2	$*$ 14.8
SVM (from [13])	5.4	25.4	30.1	20.3
GMM *ML*	**5.5**	44.6	26.0	26.6
GMM *init*	**5.5**	45.0	25.8	26.5
GMM *adapt*	6.4	**25.6**	**22.8**	**19.4**
GMMext *ML*	5.6	38.8	21.3	23.9
GMMext *init*	**5.1**	37.2	21.2	23.8
GMMext *adapt*	6.2	**23.7**	**17.6**	**18.6**
1D-HMM *ML*	$*$ **2.4**	26.6	21.8	21.6
1D-HMM *init*	5.1	27.4	21.8	21.9
1D-HMM *adapt*	6.9	**16.0**	**17.3**	**19.8**
P2D-HMM *ML*	8.3	27.0	23.0	22.1
P2D-HMM *init*	10.1	25.5	22.6	22.0
P2D-HMM *adapt*	**3.4**	$*$ **12.7**	$*$ **15.4**	**16.4**

Table 2. HTER performance of GMM (standard DCTmod2 features), GMMext (extended DCTmod2 features), 1D HMM and P2D HMM. *ML*: client models trained using traditional ML criterion; *init*: client models trained using ML initialized with a generic model; *adapt*: client models trained using MAP. The asterix indicates the best result for a protocol, while boldface indicates the best result within a model type and protocol.

approach is not better than 1D HMM; this can be explained by the much larger number of parameters used in P2D HMM (hence requiring more training data). However, when MAP training is used, the lack of data problem is effectively circumvented, resulting in the P2D HMM approach obtaining (in almost all cases) significantly better performances than the other generative models presented in this paper; moreover, in three out of four cases, the P2D HMM system performs better than the LDA/NC system presented in [13].

6. Conclusions and Future Work

It has been shown previously that systems based on local features and relatively complex generative models, namely 1D Hidden Markov Models (HMMs) and pseudo-2D HMMs, are suitable for face recognition. Recently a simpler generative model, namely the Gaussian Mixture Model (GMM), was also shown to perform well.

In this paper we first proposed to increase the performance of the GMM approach (without sacrificing its simplicity) through the use of local features with embedded positional information; we showed that the performance obtained is comparable to 1D HMMs. Secondly, we evaluated different training techniques for both GMM and HMM based systems. We showed that the traditionally used Maximum Likelihood (ML) training approach has problems estimating robust model parameters when there is only a few training images available; we showed that models estimated with MAP training (where the lack of data problem can be effectively circumvented) are significantly more robust and are able to generalize to adverse conditions present in the English corpus of the BANCA database; further experiments on the French corpus can be performed to validate these results.

While in this work we did not take into account automatic face localization, we note that techniques based on holistic representation (which in effect rigidly preserve spatial relations between facial characteristics), can be adversely affected by incorrect face localization [2]; this is in contrast to local feature based approaches (such as GMMs and HMMs) where the spatial relation between facial characteristics is less constrained; in future work we will hence evaluate the robustness of GMMext and P2D HMM approaches to imperfect face localization. In other future work we will investigate effects of embedding positional information into feature vectors used in the P2D HMM approach; moreover, we will examine why different overlap settings in DCTmod2 feature extraction are preferred by different models.

Acknowledgments

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 supported by the European projects BANCA and CIMWOS, through the Swiss Federal Office for Education and Science (OFES). The authors also thank S. Marcel for useful suggestions. The GMM and HMM systems were implemented with the aid of the *Torch* machine learning library [3].

References

[1] E. Bailly-Baillière, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Porée, B. Ruiz, J.-P. Thiran, "The BANCA Database and Evaluation Protocol", *Proc. Audio- and Video-Based Biometric Person Authentication*, Guilford, 2003, pp. 625-638.

[2] F. Cardinaux, C. Sanderson, S. Marcel, "Comparison of MLP and GMM Classifiers for Face Verification on XM2VTS", *Proc. Audio- and Video-Based Biometric Person Authentication*, Guilford, 2003, pp. 911-920.

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

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

[5] R. Duda, P. Hart, G. Stork, *Pattern Classification*, Wiley, 2001.

[6] S. Eickeler, S. Müller, R. Gerhard, "Recognition of JPEG Compressed Face Images Based on Statistical Methods", *Image and Vision Computing*, 18 (4), 2000, 279-287.

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

[8] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, Addison-Wesley, 1993.

[9] J. Mariéthoz and S. Bengio, "A Comparative Study of Adaptation Methods for Speaker Verification", *Proc. Int. Conf. Spoken Language Processing*, Denver, 2002, pp. 581-584.

[10] A. Nefian and M. Hayes, "Face recognition using an embedded HMM", *Proc. Audio and Video-based Biometric Person Authentication*, Washington D.C., 1999, pp. 19-24.

[11] A. Nefian and M. Hayes, "Maximum likelihood training of the embedded HMM for face detection and recognition", *Proc. IEEE Int. Conf. Image Processing*, Vancouver, 2000, Vol. 1, pp. 33-36.

[12] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", in: *Readings in Speech Recognition* (eds.: A. Waibel and K.-F. Lee), Kaufmann, San Mateo, 1990, pp. 267-296.

[13] M. Sadeghi, J. Kittler, A. Kostin, K. Messer, "A Comparative Study of Automatic Face Verification Algorithms on the BANCA Database", *Proc. Audio- and Video-Based Biometric Person Authentication*, Guilford, 2003, pp. 35-43.

[14] F. Samaria, *Face Recognition Using Hidden Markov Models*, PhD Thesis, University of Cambridge, 1994.

[15] C. Sanderson and K.K. Paliwal, "Fast Features for Face Authentication Under Illumination Direction Changes", *Pattern Recognition Letters*, 24 (14), 2003, 2409-2419.

[16] C. Sanderson and S. Bengio, "Statistical Transformation Techniques for Face Verification Using Faces Rotated in Depth", IDIAP Research Report 04-04, Martigny, Switzerland, 2004. *(see www.idiap.ch)*

[17] J. Zhang, Y. Yan, M. Lades, "Face recognition: Eigenfaces, elastic matching, and neural nets", *Proc. IEEE*, 85 (9), 1997, 1422-1435.