IDIAP RESEARCH REPORT

ON-LINE UNSUPERVISED ADAPTATION FOR FACE VERIFICATION USING GAUSSIAN MIXTURE MODELS WITH MULTIPLE USER MODELS

Niklas Johansson   Chris McCool   Sébastien Marcel

Idiap-RR-07-2011

MARCH 2011

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

On-line unsupervised adaptation for face verification using Gaussian Mixture Models with multiple user models

***Abstract*— In this paper, we present an initial study of on-line unsupervised adaptation for face verification. To the authors' knowledge this is the first study of this type. The key contributions consist of four test scenarios for the BANCA database as well as two novel adaptation strategies that use multiple user models. We show that by using multiple user models for each user, we can perform on-line unsupervised adaptation with a consistent increase in verification performance. Finally, we show that one of the proposed strategies performs better, or as well as the baseline, in four test scenarios at all adaptation thresholds evaluated. This suggests that this strategy is robust against both changing conditions and inexact thresholds.**

I. INTRODUCTION

Face verification is the task of authenticating the identity of a person using his or her face. The face is appealing for biometric verification because it can be acquired in a non-intrusive way (without the interaction of the user). Unfortunately, because of the changing character of the face (in different lighting conditions, with pose and over time) it is sometimes challenging to achieve robust performance [1]. In this paper we investigate unsupervised adaptation to improve the performance of a face verification system by providing diverse and more enrolment samples.

It has been shown that the use of more samples during enrolment can improve the performance of both a face and a speaker verification system [2][3]. In addition, using samples from different conditions and sessions increases the performance in speaker verification [4]. One way to increase the number and the diversity of samples is to continuously incorporate test samples after they have become available. This is referred to as *continuous unsupervised adaptation* [5].

There are two main types of unsupervised adaptation: on-line and off-line adaptation[5]. In *on-line* adaptation, the decision to use a sample for adaptation is taken directly after the sample has been seen. In contrast, in *off-line* adaptation we aggregate samples before the adaptation procedure. In this initial evaluation for face verification we focus only on on-line adaptation. The advantage of on-line adaptation is that it is faster, easier to implement and requires less storage. Its drawback is that it cannot propagate confidence between samples.

The objective of this paper is to evaluate continuous on-line unsupervised adaptation for face verification. To the authors' knowledge no previous work has been performed on continuous unsupervised adaptation for face verification. The two key contributions in this paper are the proposed test scenarios for the BANCA database in Section V and two novel multi-model adaptation strategies in Section III. We start by describing face verification and model adaptation in the framework of Gaussian Mixture Models.

II. FACE VERIFICATION USING GAUSSIAN MIXTURE MODELS

In face verification a template is created using a collection of reference images containing the user's face. We can then authenticate the user by measuring how similar a new instance of the user's face is to this template.

For this work we chose to use the Gaussian mixture model (GMM) technique for face verification proposed by Cardinaux et al. [6]. We chose this technique because it has been used by several researchers [7], [6], [8] to perform face verification and it has well defined and derived formulas for performing adaptation by using maximum a posteriori (MAP) adaptation [9].

Generally for Gaussian Mixture Models (GMMs)[6][9] we require two set of reference images. The first set contains face-images that stem from many different people. Whereas the second set only contains face-images of the target user. Loosely speaking for GMMs we use the first set to model what a general "face" looks like; referred to as the *world model (ω_{world})*. Whereas the second set is used to model the face of a particular user, called the *user model (ω_{user})*.

For GMMs we calculate the *log likelihood ratio* for a unknown input sample x as follows [9]:

$$\Lambda(x|\omega_{user}) = log\ p(x|\omega_{user}) - log\ p(x|\omega_{world}), \qquad (1)$$

where Λ referred to as the *score* and is used to predict if the sample belongs to the user or not. In our system the decision to authenticate a user for a given sample x is taken by a *threshold* ν

$$\begin{cases} \Lambda(x|\omega_{user}) > \nu, & accept \\ \Lambda(x|\omega_{user}) \leq \nu, & deny \end{cases} \qquad (2)$$

where ν is a global threshold set the same for all users.

After this short introduction to face verification using GMMs we present model adaption. For a full presentation of GMMs for verification please see [6] or [9].

III. MODEL ADAPTATION

We start by presenting how to trigger adaptation using an adaptation threshold. Thereafter we present and motivate the use of multiple models for each user.

A. Adaptation threshold

Even if there is no previous work for face verification, there do exist studies for speaker verification on on-line unsupervised adaptation. Two common techniques in speaker verification are *adaptation thresholds* and *confidence measure based model adaptation*[3][10][11]. Confidence measure base adaptation uses the score from the classifier to derive a weight for each test sample. All samples are "used" for adaptation but those with a low likelihood ratio (score) have less weight. However, it is non-trivial to derive a well-suited function that maps a weight to each sample. Therefore for this initial study we only used the method of an adaptation threshold to trigger adaptation.

The score Λ for a given sample x is compared to a fixed threshold τ. If the score is above the threshold, the sample is used for adaptation:

$$\begin{cases} \Lambda(x|\omega_{user}) > \tau, & adapt \\ \Lambda(x|\omega_{user}) \leq \tau, & skip \end{cases} \tag{3}$$

The drawback of an adaptation threshold is that when an impostor sample scores above the threshold it is equally important as a correct sample. This can be seen as a corruption of a user model. There is therefore a trade-off between accepting new samples (which is the purpose of adaptation) and excluding impostor samples (preventing corruption).

B. Motivation and use of multiple models

We assume that no adaptive system is perfect and it will falsely accept impostor samples for adaptation. It is therefore important for the system to be robust when it falsely accepts impostor samples. To handle this we propose a system that has multiple models for each user. One advantage of such a system is when an impostor samples get incorporated, it will only affect one model at the time. In addition a mixture of experts can increase performance in some circumstances [12]. Some of the drawbacks of the proposed system is that it is slower for testing and that it use more memory.

The problem of scoring against a set of multiple classifiers (models) was covered by Kittler et al. [12]. The authors concluded that the *sum rule* performed better than the other evaluated methods (min, max, product). However, because we might have different number of models we modify the sum rule to the average rule. We define the score for a sample x given a collection of models Ω as follows:

$$\Lambda(x|\Omega) = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left(\Lambda(x|\omega) - \Lambda(x|\omega_{world}) \right) \tag{4}$$

where $|\Omega|$ is the number of models in our collection, $\Lambda(\mathbf{x}|\omega)$ is the score for an individual model ω in the collection and $\Lambda(\mathbf{x}|\omega_{world})$ is the score against the world model.

IV. Adaptation strategies

We now present the four different adaptation strategies that are used for this evaluation. All four strategies share the fact that a new model is created using *maximum a posteriori (MAP)* adaptation [9] from the world model. We use Ω to refer to a collection of models. For the first two strategies this collection Ω only contains one single model. The last two strategies are the main contributions in this paper.

The first strategy **SM** is the simplest of the four. In this strategy the new user model ω_{new} is created using only the last unsupervised test sample. The old model is then replace with the newly created one. This can more formally be written as:

SM

$$\begin{aligned} \text{Adapt using only new sample:} \quad & \omega_{world} \longrightarrow \omega_{new} \\ \text{Replace old model:} \quad & \Omega := \{\omega_{new}\} \end{aligned}$$

The second strategy **SM+** is an extension of the first. In addition to the unsupervised test sample we use all previously seen samples when adapting.

SM+

$$\begin{aligned} \text{Adapt using all samples:} \quad & \omega_{world} \longrightarrow \omega_{new} \\ \text{Replace old model:} \quad & \Omega := \{\omega_{new}\} \end{aligned}$$

In the third strategy **MM** we add the new model to the collection instead of replacing the old model. As in SM the new model is created using only the latest unsupervised test sample. This can more formally be written as:

MM

$$\begin{aligned} \text{Adapt using only new sample:} \quad & \omega_{world} \longrightarrow \omega_{new} \\ \text{Add model to collection:} \quad & \Omega := \Omega \cup \{\omega_{new}\} \end{aligned}$$

The final strategy **MM+** is a modification of the previous strategy. To create the new model we use the unsupervised test samples plus additional samples. These additional samples are the same samples that were used to create ω^*, the best scoring model during test of sample x. The **MM+** strategy can be formally written as:

MM+

$$\begin{aligned} &\text{Select the model that scored the} \\ &\text{best during test of latest sample:} \quad && \omega^* = \operatorname*{argmax}_{\omega \in \Omega} \Lambda(x|\omega) \\ &\text{Recall samples used to created } \omega^*: && y \\ &\text{Adapt using new samples and } y: && \omega_{world} \longrightarrow \omega_{new} \\ &\text{Add the model to the collection:} && \Omega := \Omega \cup \{\omega_{new}\} \end{aligned}$$

V. Protocols for on-line unsupervised adaptation

We start by presenting the BANCA database and the three conditions. Thereafter we present the proposed test scenarios (protocols).

Fig. 1. Examples of the three conditions in the BANCA database.

A. Database

We used the BANCA (English) database for our experiments. The BANCA database is suitable because each user is captured in three different conditions. The database consists of 26 females and 26 males [13]. Each user is captured in three different conditions with four sessions in each condition. The first four sessions belongs to the *Controlled* condition, the next four to the *Degraded* condition and the final four sessions belongs to the *Adverse* condition. An example of the three conditions is given in Figure 1.

B. Proposed protocols for on-line adaptation

Four different testing scenarios (protocols) were created to evaluate our on-line adaptation strategies. They are described in detail below but the two key aspects we wanted to investigate were the effect of:

1) the composition of the testing environments, and
2) the order of client and impostor trials.

We based our *on-line adaptation* protocols on BANCA protocol P. According to this protocol, we first train the user model with session 1 (controlled condition). Then we test on sessions $2, 3, 4$ (also controlled conditions). Thereafter we test on sessions $6, 7, 8$ (degraded conditions) and finally we test on sessions $10, 11, 12$ (adverse conditions).

Normally the order of the testing sessions does not matter. However for an adapting system the order is important and it can change the end result. Therefore we created the "normal" testing order and the "mixed" testing order. For the "normal" testing order we tested in increasing session order. This means that we test our system in one condition at a time, first all controlled then all degraded and finally all adverse. For the "mixed" testing order we cycled over the testing condition, therefore each client test was performed in a different condition from the previous one. For the "normal" test order we shift conditions only two times, between 4 and 6 and between 8 and 10. Whereas in the "mixed" test order we shift conditions eight times. These two testing orders are presented in detail in Table II and III.

Two additional protocols were derived from the first two. Their purpose is to evaluate the effect of seeing impostor trials before client trials in each session. We changed the protocols in Table II and III so that the impostor accesses occurred first in each session. This gives us four different testing scenarios summarised in Table I.

VI. Experiments

We used a state-of-the-art face detector to locate the faces in all of the videos; one face per frame. We cropped each

TABLE I

SUMMARY OF THE FOUR DIFFERENT ON-LINE ADAPTING PROTOCOLS USED FOR OUR EXPERIMENTS.

	Session order	Access order	Short name
1.	Conditions in numeric order	Client first	norm
2.	Conditions in mixed order	Client first	mixed
3.	Conditions in numeric order	Impostor first	norm rev
4.	Conditions in mixed order	Impostor first	mixed rev

TABLE II

ONE OF THE FOUR ON-LINE ADAPTATION PROTOCOLS. WE CALL THIS ORDER OF THE SESSIONS "NORMAL". IN THIS PROTOCOLS THE CLIENTS ARE TESTED AHEAD OF THE IMPOSTORS IN EACH SESSION. WHEN CHANGING THE ORDER OF CLIENT AND IMPOSTOR ATTEMPTS WE REFER TO THIS PROTOCOL AS "NORMAL REVERSED".

Step	Task	Using session	Condition
1	Enrol	1	Controlled
2	Test client access	2	Controlled
3	Test impostor access	2	Controlled
4	Test client access	3	Controlled
5	Test impostor access	3	Controlled
6	Test client access	4	Controlled
7	Test impostor access	4	Controlled
8	Test client access	6	Degraded
9	Test impostor access	6	Degraded
10	Test client access	7	Degraded
11	Test impostor access	7	Degraded
12	Test client access	8	Degraded
13	Test impostor access	8	Degraded
14	Test client access	10	Adverse
15	Test impostor access	10	Adverse
16	Test client access	11	Adverse
17	Test impostor access	11	Adverse
18	Test client access	12	Adverse
19	Test impostor access	12	Adverse

TABLE III

ONE OF THE FOUR ON-LINE ADAPTATION PROTOCOLS. WE CALL THIS ORDER OF THE SESSIONS "MIXED". IN THIS PROTOCOL THE CLIENTS ARE TESTED AHEAD OF THE IMPOSTORS IN EACH SESSION. WHEN CHANGING THE ORDER OF CLIENT AND IMPOSTOR ATTEMPTS WE REFER TO THIS PROTOCOL AS "MIXED REVERSED".

Step	Task	Using session	Condition
1	Enrol	1	Controlled
2	Test client access	2	Controlled
3	Test impostor access	2	Controlled
4	Test client access	6	Degraded
5	Test impostor access	6	Degraded
6	Test client access	10	Adverse
7	Test impostor access	10	Adverse
8	Test client access	3	Controlled
9	Test impostor access	3	Controlled
10	Test client access	7	Degraded
11	Test impostor access	7	Degraded
12	Test client access	11	Adverse
13	Test impostor access	11	Adverse
14	Test client access	4	Controlled
15	Test impostor access	4	Controlled
16	Test client access	8	Degraded
17	Test impostor access	8	Degraded
18	Test client access	12	Adverse
19	Test impostor access	12	Adverse

face and normalised it for size (80-by-64). We then applied the preprocessing step that is described by Tan and Triggs [15]. To make the experiments computationally feasible we only used a subset of the frames in each video. We used a subset of 30 frames per video, picked uniformly over the whole video.

In the experiments we used Gaussian Mixture Models (GMMs) to model the feature vectors [9]. Each user model was adapted from the world model. The components in the world model were initialised randomly and then clustered using the K-means algorithm (using 500 mixture components and 100 iterations). The world model was then built using the expectation-maximisation algorithm with 10 iterations on these clusters.

To create the feature vector we used the extended DCT-mod2 method presented in [6]. First, we split the cropped image (80-by-64) into 285 blocks, each block 8-by-8 pixels. Where each block was picked using a 4 pixel overlap both vertically and horizontally. Second, for each block a 2D discrete cosine transform (DCT) was applied resulting in 64 coefficients. From the 64 coefficients, we picked the "first" 15 coefficients in a zig-zag pattern as described in [7]. However, the first coefficient (bias) was completely discarded and coefficients 2 and 3 were transformed into four delta coefficients. A delta coefficient is the differences in either horizontal or vertical values compared to adjacent blocks. Finally, the row and column indices were concatenated to the feature vector. In total, the dimension of the feature vector was 20 (picked 15, discard bias, 2 coefficients are transformed into 4 delta coefficients, finally add x and y position of block).

A. Parameter settings

The adaptation threshold is central for the performance of the adaptation system. Several different thresholds were therefore chosen for evaluation. We focused on the impostor score distribution to choose the adaptation threshold. This distribution is easier to approximate because it is possible to produce more impostor samples than client samples. We chose adaptation thresholds at a fixed interval from the impostor mean using the impostor score variance as the step size. The impostor mean (μ^I) and the impostor variance (σ^I) were derived on the specified development set in the BANCA database. We evaluated five different adaptation thresholds presented in Table IV.

B. Performance Metrics

The results in this paper are measured using either *average half total error rate (HTER)* or with an *expected performance curve (EPC)* [14]. These are the usual methods for reporting results on the BANCA database.

VII. Results and discussion

A. Supervised adaptation

We start by presenting the results for the four strategies in supervised adaptation, that is, using all the true client samples and none of the impostor ones. As baseline we use a system that enrol with the same data but does not adapt.

TABLE IV

The 5 different adaptation thresholds used for on-line adaptation, where μ^I and σ^I are the mean and standard deviation of the impostor distribution (derived independently on the specified development set).

#	Adapting threshold
1	μ^I +0.5 σ^I
2	μ^I +1.0 σ^I
3	μ^I +1.5 σ^I
4	μ^I +2.0 σ^I
5	μ^I +2.5 σ^I

TABLE V

Results for supervised adaptation given in *average half total error rate*. The four adaptation strategies were evaluated in four test scenarios (protocols).

	Client first		Impostor first	
	Normal	Mixed	Normal	Mixed
MM+	**0.16**	0.16	0.16	**0.15**
SM+	**0.16**	**0.15**	**0.12**	0.16
MM	0.19	0.17	0.18	0.15
SM	0.21	0.42	0.20	0.25
Non-adapting system			0.21	

In Table V we present the results (average half total error rate) for on-line supervised adaptation. One strategy (**SM**) performs worse than baseline in three out of the four test scenarios. It is clear that the extended version of this strategy **SM+** performs significantly better. This indicates a need to keep previously seen samples when adapting.

Excluding the worst strategy (**SM**), we see an performance increase in all four test scenarios. This means that better performance can be achieved using on-line adaptation. However, the performance varies with different test scenarios. This indicates the importance of using multiple protocols when evaluating on-line supervised adaptation.

The **SM+** strategy is the best strategy and performs equal or better compared to the other four strategies in three out of the four test scenarios. This suggests that it is efficient to model all samples together.

B. Unsupervised adaptation

In Table VI we present the results (*average half total error rates*) for unsupervised on-line adaptation. The four adaptation strategies **MM+**, **SM+**, **MM** and **SM** were tested in the four different scenarios using five different adaptation thresholds.

The results suggest that it is not easier to adapt using the "normal" test order compared to the "mixed" one. One key difference between the two test orders is the number of shifts between conditions during test, two shifts compared to eight. This suggest that there is no significant penalty in performance every time we change condition.

From the variances in Table VI it seems that the two multi-model strategies (**MM** and **MM+**) perform more consistently.

TABLE VI

RESULTS FOR UNSUPERVISED ADAPTATION MEASURED IN *average half total error rate* THE FOUR STRATEGIES ARE TESTED IN FOUR DIFFERENT SCENARIOS WITH FIVE DIFFERENT ADAPTATION THRESHOLDS.

	Client first		Impostor first			
	Normal	Mixed	Normal	Mixed	Mean	Var / 10e-4
Threshold μ^I +0.5 σ^I						
MM+	**0.18**	**0.19**	**0.17**	**0.18**	0.18	**0.46**
SM+	0.24	0.22	0.18	0.19	0.21	5.53
MM	0.26	0.23	0.27	0.22	0.24	5.93
SM	0.42	0.45	0.39	0.33	0.40	24.65
Threshold μ^I +1.0 σ^I						
MM+	0.18	0.20	0.18	0.19	0.19	**1.13**
SM+	0.21	0.24	0.20	0.20	0.21	3.57
MM	0.24	0.23	0.24	0.22	0.23	**0.99**
SM	0.38	0.37	0.35	0.33	0.36	4.89
Threshold μ^I +1.5 σ^I						
MM+	0.19	0.21	0.19	0.20	0.20	**0.69**
SM+	0.24	0.23	0.20	0.21	0.22	3.61
MM	0.23	0.22	0.23	0.22	0.23	**0.25**
SM	0.32	0.29	0.31	0.29	0.30	2.36
Threshold μ^I +2.0 σ^I						
MM+	0.19	0.19	0.19	0.19	0.19	**0.02**
SM+	0.22	0.22	0.19	0.20	0.21	2.80
MM	0.22	0.22	0.22	0.22	0.22	**0.11**
SM	0.25	0.25	0.27	0.26	0.26	1.63
Threshold μ^I +2.5 σ^I						
MM+	0.19	0.19	0.19	0.19	0.19	**0.10**
SM+	0.20	0.21	0.19	0.20	0.20	**1.02**
MM	0.22	0.21	0.22	0.22	0.22	**0.15**
SM	0.24	0.24	0.25	0.26	0.25	1.37
Non-adapting system			0.21			

From the results we notice that the multi-model strategy **MM+** performs better than the other three. The best performance for the strategy is achieved using the lowest threshold (μ^I +0.5 σ^I). Moreover the strategy performs better or as well as the baseline in all four test scenarios and for all five thresholds. This suggests that it is robust against changing conditions as well as inexact thresholds.

In Table VI we notice that two out of the four strategies, **MM** and **SM**, never performs better than the baseline. In addition they perform better using higher thresholds than with lower thresholds. This is especially true for the worst strategy **SM**. This indicates that these strategies are not as useful for unsupervised adaptation.

We present an EPC plot [14] for the best strategy **MM+** in Figure 2. The thick line in the figures indicates a non-adaptive system (baseline). We notice in Figure 2 that the strategy performs significantly better than baseline around the half total error point ($a = 0.5$) for all four test scenarios.

VIII. CONCLUSION

In this paper we presented an initial study for on-line unsupervised adaptation for face verification. Because there were no previous studies, we created four new test scenarios and four new adaptation strategies. The four test scenarios and two of the four methods were the key contributions in this paper.

We found one multi-model strategies that performed better or as well as the baseline in all four test scenarios and for all adaptation thresholds evaluated. This suggest that

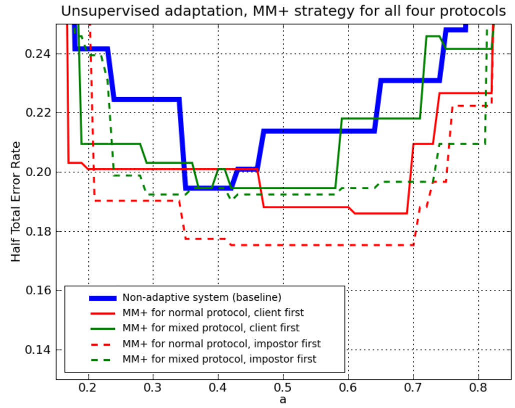

Fig. 2. **Unsupervised results for MM+** - In this figure we evaluate the **MM+** strategy for all four testing scenarios. For each protocol we have pick the best threshold. For **MM+** the best threshold is the lowest $\mu^I + 0.5\sigma^I$.

this strategy is robust against both changing conditions and inexact thresholds.

Results suggest that a single-model adaptation strategy is the best strategy for supervised adaptation on the BANCA database whereas a multi-model strategy is best in unsupervised adaptation.

In the future, we aim to expand our evaluation by using a larger database. We will create more test scenarios, primarily in which there is an imbalance between the number of clients and impostors. In addition, we will implement confidence measure based model adaption as used in speaker verification.

REFERENCES

[1] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *Acm Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[2] N. Poh and S. Bengio, “Improving Face Authenticaion using Virtual Samples,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, vol. 6, 2003, p. 10.

[3] A. Preti, J. Bonastre, D. Matrouf, F. Capman, and B. Ravera, “Confidence measure based unsupervised target model adaptation for speaker verification,” in *Proc. Interspeech*, 2007, pp. 754–757.

[4] R. Vogt and S. Sridharan, “Experiments in session variability modelling for speaker verification,” in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*, vol. 1, 2006.

[5] O. Viikki and K. Laurila, “Incremental On-line Speaker Adaptation in Adverse Conditions,” in *Fifth International Conference on Spoken Language Processing*, 1998.

[6] F. Cardinaux, C. Sanderson, and S. Bengio, “Face verification using adapted generative models,” *Automatic Face and Gesture Recognition, IEEE International Conference on*, vol. 0, p. 825, 2004.

[7] C. Sanderson and K. Paliwal, “Polynomial features for robust face authentication,” in *Proceedings of International Conference on Image Processing*, vol. 3. Citeseer, 2002, pp. 997–1000.

[8] S. Lucey and T. Chen, “A GMM parts based face representation for improved verification through relevance adaptation,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004.

[9] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, January 2000. [Online]. Available: http://dx.doi.org/10.1006/dspr.1999.0361

[10] C. Barras, S. Meignier, and J. Gauvain, “Unsupervised online adaptation for speaker verification over the telephone,” in *ODYSSEY04-The Speaker and Language Recognition Workshop*. Citeseer, 2004.

[11] N. Mirghafori and L. Heck, “An adaptive speaker verification system with speaker dependent a priori decision thresholds,” in *Seventh International Conference on Spoken Language Processing*, 2002.

[12] J. Kittler, M. Hatef, R. Duin, and J. Matas, “On combining classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[13] E. Bailly-Bailliere, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Porée *et al.*, “The BANCA database and evaluation protocol,” in *Audio-and Video-Based Biometric Person Authentication*. Springer, 2003, pp. 1057–1057.

[14] S. Bengio and J. Mariéthoz, “The expected performance curve: a new assessment measure for person authentication,” in *ODYSSEY04-The Speaker and Language Recognition Workshop*. Citeseer, 2004.

[15] X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” in *Proceedings of the 3rd international conference on Analysis and modeling of faces and gestures*. Springer-Verlag, 2007, pp. 168–182.