SPEECH & FACE BASED BIOMETRIC AUTHENTICATION AT IDIAP

Conrad Sanderson, Samy Bengio, Herve Bourlard, Johnny Mariéthoz,
Ronan Collobert, Mohamed F. BenZeghiba, Fabien Cardinaux, Sébastien Marcel *

Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)
Rue du Simplon 4, CH-1920 Martigny, Switzerland
www.idiap.ch

ABSTRACT

We present an overview of recent research at IDIAP on speech & face based biometric authentication. This paper covers user-customised passwords, adaptation techniques, confidence measures (for use in fusion of audio & visual scores), face verification in difficult image conditions, as well as other related research issues. We also overview the open source Torch library, which has aided in the implementation of the above mentioned techniques.

1. INTRODUCTION

The goal of a biometric identity verification (authentication) system is to either accept or reject the identity claimed by a given person, based on the person's characteristics such as speech, face or fingerprints. Applications range from access control, transaction authentication (e.g. telephone banking), voice mail, secure teleworking, to forensic work, where the task is to determine whether a biometric sample belongs to a given suspect [12].

In this paper we present an overview of recent research at IDIAP in the fields of speaker verification (Section 2), face verification (Section 3) and multi-modal verification (Section 4). In Section 5 we describe an open source machine-learning library, called Torch, which has aided in the implementation of the above mentioned techniques.

As a thorough introduction to the field of biometrics is beyond the scope of this paper, it is assumed that the reader is familiar with basic concepts in speaker, face and multi-modal verification. Recent introductory and review material can be found in [5, 14, 27].

2. SPEAKER VERIFICATION

2.1. Comparison of Several Adaptation Methods

Gaussian Mixture Models (GMMs), the main tool used in text-independent speaker verification [25], can be trained using the Expectation Maximization (EM) algorithm [11]. However, in order to obtain correctly estimated models, large amount of training data for each client is generally needed, which is usually difficult to obtain in real applications. Hence several adaptation methods, which start from a general model and adapt it for specific clients, have been proposed in order to overcome this problem. We recently compared [23] some of them in order to assess their relative performance on the NIST database [12]. We compared the classical Bayesian Maximum a Posteriori (MAP) principle [17] with two other techniques, Maximum Likelihood Linear Regression (MLLR) [16] and eigenvoices [20] (inspired by eigenfaces [30]). Table 1 shows that the simple MAP technique is still the best adaptation method for GMM-based speaker verification.

One explanation for the poor results of MLLR and EigenVoices might be that both methods force the parameters of the client models to be in a smaller parameter space, defined by training clients (previously seen but not used during testing); this may be good for discriminating clients from everything else, but not necessarily good for discriminating clients from each other.

Method	ML	MAP	MLLR	Eigen
HTER[1]	22.9	15.8	18.42	20.57

Table 1. Performance of adaptation methods on the NIST database.

2.2. Synchronous Alignment

Classical text-dependent speaker verification systems are based on two Hidden Markov Models (HMMs): the client model θ_{client} and the anti-client model (world model) θ_{world}. The Viterbi algorithm is then used to find the best path through these models. The main idea of synchronous alignment is to force this path to be the same for the two models. In [24] we proposed a new Viterbi criterion for such a task:

$$Q^* = \arg\max_Q p(X, Q|\theta_{client})^{(1-\alpha)} \cdot p(X, Q|\theta_{world})^{\alpha}$$

where Q^* is the best estimated path, X are the observations and α the weight given to the world model.

A similar approach can be used in text-independent speaker verification using GMMs: we force the Gaussians that maximize the likelihood of the observations to be the same in the two models; initial results show that this approach is more robust in difficult conditions (poor client data, noisy data) and simplifies the mathematics.

2.3. Decision Strategies

The statistical framework used in speaker verification usually involves the estimation of the log likelihood ratio of the access given the client and world models. This ratio is then compared to a threshold which should in theory be equal to 0, when no other priors are available. In practical applications, this threshold is in fact estimated on a separate development set in order to reach the Equal Error Rate (EER) or to minimize HTER[1]. Instead of searching for such a threshold, we proposed in [2] to estimate a more complex function of the obtained average log likelihoods given the client and world models. We compared several approaches such as Multi-Layer Perceptrons (MLPs) and Support Vector Machines (SVMs). On the PolyVar database (over 36,000 tests), the HTER was reduced from 5.55% (using the standard threshold) to 4.73% (using the SVM decision approach).

*The authors thank the Swiss National Science Foundation for supporting this work through the National Centre 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).

[1] Half Total Error Rate (HTER) is defined as $\frac{1}{2}$(FA%+FR%), where FA% is the false acceptance rate and FR% is the false rejection rate.

2.4. User-Customised Passwords

In a typical text-dependent speaker verification system, the speaker is constrained to a single phrase or a set of words for which the system has *a priori* knowledge (e.g. correct phonetic transcription of the phrase, or the vocabulary from which the phrase can be chosen is very limited [e.g. 10 digits]). Compared to text-independent systems, where the user can utter any text, text-dependent systems are less user-friendly but generally have better discrimination ability. In User-Customised Password (UCP) systems [29], the system does not place any constraints on the password: users are free to choose any text.

Implementation of a UCP system raises several issues; first, we have to infer the HMM topology of the password; second, we have to create (using adaptation techniques) a speaker dependent model which models both the lexical content of the password as well as the speaker's characteristics. Formally, a speaker pronouncing utterance X is classified as a true claimant S_k associated with password M_k when:

$$P(M_k, S_k|X) \geq P(M_k, \overline{S}_k|X) \quad (1)$$

$$\text{and} \quad P(M_k, S_k|X) \geq P(\overline{M}_k, S|X) \quad (2)$$

where $P(M_k, S_k|X)$, $P(M_k, \overline{S}_k|X)$ and $P(\overline{M}_k, S|X)$, are, respectively, the joint posterior probability of a true client pronouncing the correct password, an impostor pronouncing the correct password and any speaker pronouncing any other password.

From the above decision rules we have derived two approaches, described in Sections 2.4.1 & 2.4.2. Both approaches use the same phonetic inference technique, described as follows: a hybrid HMM/ANN[2] system [6] is used to infer the phonetic transcription for each repetition of the password; based on the best phonetic transcription (yielding the highest normalised posterior probability), the topology of the HMM password M_k is selected.

2.4.1. HMM based

Using Bayes rule, decision rules (1) and (2) can be rewritten as follows[3] [3]:

$$\frac{P(X|M_k, S_k)}{P(X|M_k, \overline{S}_k)} \leq \delta_1 \quad (3)$$

$$\text{and} \quad \frac{P(X|M_k, S_k)}{P(X|\overline{M}_k, S)} \leq \delta_2 \quad (4)$$

The terms on the left side of Eqns. (3) & (4) can be interpreted, respectively, as the speaker verification score (when the speaker pronounces the correct password) and the utterance verification score. A weighted sum combination technique is used to estimate the final score [27]. In this approach we adapt (using speaker's training data and MAP adaptation) the inferred HMM password M_k, in which each state is a phoneme modeled by a 3-state HMM model with 3 Gaussians per state. This approach will be referred to as *SYS-A*.

2.4.2. Combined HMM/ANN and GMM based

Using the conditional probability rule, decision rules (1) and (2) can be rewritten as follows [4]:

$$\left[\frac{P(M_k|S_k, X)}{P(M_k|\overline{S}_k, X)}\right]\left[\frac{P(X|S_k)}{P(X|\overline{S}_k)}\right] \geq \delta_3 \quad (5)$$

$$\left[\frac{P(M_k|S_k, X)}{P(\overline{M}_k|S, X)}\right]\left[\frac{P(X|S_k)}{P(X|S)}\right] \geq \delta_4 \quad (6)$$

[2] ANN = Artificial Neural Network

[3] Assuming that the *a priori* simultaneous probability of any speaker and any word is equal for all combinations of speakers and words

The first term in both decision rules is the *posterior probability* that the pronounced word X is M_k; it is estimated by an ANN. The second term is the verification score found using a text-independent GMM-based system. A weighted sum combination technique is used to combine the two scores. For each speaker we adapt a single-layer perceptron and a GMM. We shall refer to this approach as *SYS-B*.

2.4.3. Evaluation

Results on the PolyVar Database [9], using both inferred and correct phonetic transcriptions, are shown in Table 2. We can see that *SYS-A* is somewhat sensitive to the accuracy of the transcription process. For *SYS-B* we have found that the performance is close to using the GMM sub-system alone, indicating that when a GMM model is trained using only short words, it becomes speaker- as well as speech-dependent.

	SYS-A (I)	*SYS-A* (C)	*SYS-B* (I)	*SYS-B* (C)
α	0.6	0.6	0.3	0.5
EER	3.35%	3.03%	3.51%	3.45%

Table 2. Performance with optimal combination parameter α. (C) and (I) denote systems using the correct and the inferred phonetic transcription, respectively.

2.5. Future Work

In text-independent systems, verification approaches directly based on discriminative techniques such as MLPs and SVMs currently fail to match the verification performance of the (generative) GMM approach. Why is it so? One of the reasons could be the criterion used during training: MLPs and SVMs try to minimize the total classification error instead of the HTER or EER. Initial results for MLPs and SVMs trained using a more appropriate criterion are promising.

3. FACE VERIFICATION

Generally speaking, a full face verification system can be thought of as being comprised of three stages:

1. Face localisation and segmentation
2. Normalisation
3. The actual face verification, which can be further subdivided into:
   (a) Feature extraction
   (b) Classification

The second stage (normalisation) usually involves a geometric transformation (to correct for size and rotation), but it can also involve an illumination normalisation (however, illumination normalisation may not be necessary if the feature extraction method is robust against varying illumination). Here we concentrate on stage (3).

3.1. Enhanced PCA Feature Extraction

A major source of errors is the sensitivity of the feature extraction stage to illumination direction changes. While this sensitivity is a large concern in security systems, in forensic applications [21] other types of image corruption can be important; here, face images may be obtained in various illumination conditions from various sources: digitally stored video, possibly damaged and/or low quality analogue video tape or TV signal corrupted with "static" noise (see Fig. 1 for example images).

In standard Principal Component Analysis (PCA) based feature extraction (also known as eigenfaces [30]), a given face image is represented by matrix F containing grey level pixel values; F is converted to a face vector, $\vec{f}$, by concatenating all the columns; a D-dimensional feature vector, $\vec{x}$, is then obtained by:

$$\vec{x} = \mathbf{U}^T(\vec{f} - \vec{f}_\mu) \quad (7)$$

where $\mathbf{U}$ contains D eigenvectors (with largest corresponding eigenvalues) of the training data covariance matrix, and $\vec{f}_\mu$ is the mean of training face vectors.

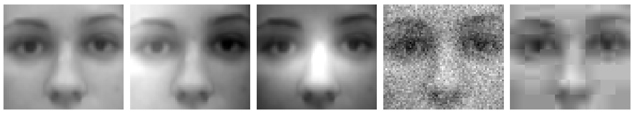

Fig. 1. Left to right: original image, corrupted with linear illumination change, Gaussian illumination change, white Gaussian noise, compression artefacts.

PCA derived features have been shown to be sensitive to changes in the illumination direction causing rapid degradation in verification performance [28]. In the proposed *enhanced PCA* approach, a given face image is processed using recently proposed *DCT-mod2* feature extraction [28] to produce pseudo-image $\hat{F}$, which is then used in place of F by traditional PCA feature extraction. Since *DCT-mod2* feature vectors are robust to illumination changes, features obtained via the *enhanced PCA* should also be robust to illumination changes. Formally, the pseudo image is constructed as follows:

$$\hat{F} = \begin{bmatrix} \vec{c}^{\,(\Delta b,\Delta a)} & \vec{c}^{\,(\Delta b,2\Delta a)} & \vec{c}^{\,(\Delta b,3\Delta a)} & \cdots \\ \vec{c}^{\,(2\Delta b,\Delta a)} & \vec{c}^{\,(2\Delta b,2\Delta a)} & \vec{c}^{\,(2\Delta b,3\Delta a)} & \cdots \\ \vec{c}^{\,(3\Delta b,\Delta a)} & \vec{c}^{\,(3\Delta b,2\Delta a)} & \vec{c}^{\,(3\Delta b,3\Delta a)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (8)$$

where $\vec{c}^{\,(n\Delta b,n\Delta a)}$ denotes the *DCT-mod2* feature vector for an 8×8 block located at $(n\Delta b, n\Delta a)$, while Δb and Δa are block location advancement constants for rows and columns respectively (here, Δb=Δa=4).

Experiments [26] on the VidTIMIT database show (see Table 3) that the *enhanced PCA* technique retains all the positive aspects of traditional PCA (that is robustness against white noise and compression artefacts) while also being robust to illumination direction changes; moreover *enhanced PCA* outperforms histogram equalisation pre-processing.

Type	*clean*	*lin. illum.*	*Gaus. illum.*	*white noise*	*compr.*
standard	3.57	27.14	32.19	3.57	3.57
hist. equ.	4.29	32.86	36.34	7.14	4.33
enhanced	5.31	7.14	18.57	5.67	6.03

Table 3. EER Performance of PCA based feature extraction

3.2. Comparison between GMM and MLP classifiers

The choice of the classifier not only has an impact on the discrimination ability of the system, but also its robustness to imperfectly located faces. Experiments on the XM2VTS database show that (when using *DCT-mod2* features [28]) the GMM approach easily outperforms the MLP approach for high resolution faces and is significantly more robust to imperfectly located faces (see Table 4). Further experiments [8] have shown that the computational requirements of the GMM approach can be significantly smaller than the MLP approach at a cost of small loss of performance.

Model type (face size)	*FA%*	*FR%*	*HTER*
GMM (80×64)	1.95	2.75	2.35
MLP (80×64)	11.55	11.25	11.40
GMM (40×32)	5.47	6.25	5.86
MLP (40×32)	7.98	9.75	8.86

Table 4. Comparison of GMM and MLP performance using automatically located faces (XM2VTS, Config. I)

4. CONFIDENCE MEASURES FOR FUSION

Several recent contributions have shown that combining the decisions or scores coming from various unimodal verification systems (based, for instance, on the voice or the face of a person) often enhances the overall authentication performance (e.g. [19, 27]). This has been shown to be true using various fusion algorithms, from the simplest ones such as product or sum rules, to the more complex ones such as SVMs or MLPs.

Various researchers and practitioners have expressed an interest in the estimation of some sort of confidence on decisions taken by authentication systems. Based on this interest, we recently analysed several methods to improve fusion algorithms by trying to estimate complementary information such as a confidence on the decision of each unimodal system [1]. One can think of the fusion algorithms as a way to somehow *weight* the scores of different unimodal verification systems, eventually in a nonlinear way, in order to give a better estimation of the overall score. If one had access not only to the scores but also to a confidence measure on these scores, this measure could help in the fusion process. Hence, intuitively, if for some reason one unimodal verification system was able to say that its score for a given access was not very precise, while a second unimodal verification system was more confident on its own score, the fusion algorithm should be able to provide a better decision than without this knowledge.

The methods proposed in [1] where rather simple. The first one was based on the hypothesis that scores coming from unimodal verification systems could have been generated by two Gaussian distributions, one for the genuine accesses and one for the impostor accesses. Based on this hypothesis, a simple confidence score can be derived. Since this Gaussian hypothesis is false in general, the second proposed method was based instead on a simple non-parametric idea: estimate the confidence associated with a score using a simple histogram. Finally, the third proposed method was based on the possibility of estimating the gradient of a simple confidence measure (such as the likelihood) that could be extracted from the model, with respect to all its parameters. The amplitude of such gradient would then give an idea of the adequacy of the model to explain the decision (a small value would mean that the model is confident, while a large value would imply a small confidence on the decision).

In experiments on the XM2VTS database [22], the above methods were used to compute additional inputs given to the fusion algorithm. Results are presented in Table 5. The traditional fusion algorithm (SVM in this case) was trained with two inputs: the log likelihood of the score given the client model and the log likelihood of the score given the world model. The "fusion + confidence" model was also an SVM, trained with four inputs: the two log likelihoods plus the two corresponding *model adequacy estimates* of the confidence of each model. While it is clear that the fusion algorithm clearly enhances the performance, adding some confidence information adds a modest relative improvement of 6% on the overall performance.

A probably more interesting way of using confidence values for authentication systems is to propose to delay (or hand over to a human) a decision when the associated confidence is lower than a given threshold. Using the non-parametric method of computing the confidence values, and selecting for instance the threshold in such a way that less than 0.64% of accesses were set aside, it was possible to reduce the overall HTER obtained on configuration I of XM2VTS from 0.69% to 0.45%, a 35% relative performance improvement.

System	*HTER*	
	Config. I	Config. II
Face only (with MLPs)	3.22	2.61
Voice only (with GMMs)	1.91	1.75
Fusion using SVMs	0.69	0.30
Fusion + Confidence	0.67	0.26

Table 5. Verification performance on XM2VTS

5. THE OPEN-SOURCE TORCH LIBRARY

The open source C++ Torch library[4] implements most state-of-the-art machine learning algorithms in a unified framework. The objective is to ease the comparison between algorithms, simplify the process of extending them and provide a platform for easy implementation of new algorithms. Unlike programs such as *Matlab* which are more suited for prototyping and toy problems, C++ programs written with the aid of Torch are able to deal with large real-life problems.

Torch can handle both static and dynamic problems. For example, Torch can deal with all kinds of "gradient-machines" which can be trained with the back-propagation algorithm [13]. Many modules are available, which can be connected with each other in order to obtain the desired machine. Creating a multi-layered perceptron, a mixture of experts, a radial basis function neural network, or even a time delay neural network or a complex convolutional neural network (spatial or temporal), takes only a few lines of C++ code with the aid of Torch.

Support Vector Machines (SVMs) [13, 31] are available in Torch; in fact, their implementation is one of the fastest available [18, 10]. Gaussian Mixture Models (GMMs), often used to represent any static distribution, have also been implemented in Torch.

The Hidden Markov Model (HMM) approach [13] is one of the most widely used techniques to represent sequences (such as biological sequences, speech data, or handwritten data). In Torch the user has the possibility to create HMMs with many kinds of distribution models, including methods based on artificial neural networks. It is also possible to train them either with an Expectation Maximization algorithm [11], with a Viterbi [32] algorithm, or even using gradient ascent. Moreover, several classes have also been implemented in order to be able to solve connected word speech recognition tasks. Small and large vocabulary decoders, compatible with Torch, are available. We have also implemented the Maximum a Posteriori (MAP) [17, 25] adaptation technique for both GMMs and HMMs.

Simple algorithms such as k-means, k-nearest neighbours or Parzen windows are provided as well. Bagging [7] and boosting [15] which are both "ensemble" algorithms, can be applied in Torch to almost any machine learning algorithm.

Being able to use all these algorithms in a simple yet unified framework enables researchers to compare them and easily enhance them. We strongly believe that providing such a platform to the community helps researchers to propose, develop and share novel algorithms more quickly.

6. REFERENCES

[1] S. Bengio, C. Marcel, S. Marcel and J. Mariéthoz, "Confidence Measures for Multimodal Identity Verification", *Information Fusion*, Vol. 3, No. 4, 2002, pp. 267-276.

[2] S. Bengio and J. Mariéthoz, "Learning the Decision Function for Speaker Verification", *Proc. ICASSP*, Salt Lake City, 2001, pp. 425-428.

[3] M. F. BenZeghiba and H. Bourlard, "User-Customized Password HMM based Speaker Verification", *Proc. COST-275 Workshop on The Advent of Biometrics on the Internet*, Rome, 2002, pp. 103-106.

[4] M. F. BenZeghiba and H. Bourlard, "Hybrid HMM/ANN and GMM Combination for User-Customized Password Speaker Verification", *Proc. ICASSP*, Hong-Kong, 2003.

[5] R.M. Bolle, J.H. Connell and N.K. Ratha, "Biometric perils and patches", *Pattern Recognition* Vol. 35, No. 12, 2002, pp. 2727-2738.

[6] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[7] L. Breiman, "Bagging Predictors", *Machine Learning*, Vol. 24, No. 2, 1994, pp. 123-140.

[8] F. Cardinaux, C. Sanderson and S. Marcel, "Comparison of MLP and GMM Classifiers for Face Verification on XM2VTS", *IDIAP-RR*** *03-10*, 2003.

[9] G. Chollet, J.-L. Cochard, A. Constantinescu, C. Jaboulet and P. Langlais, "Swiss French PolyPhone and PolyVar: telephone speech databases to model inter- and intra-speaker variability", *IDIAP-RR 96-01*, 1996.

[10] R. Collobert and S. Bengio, "SVMTorch: Support Vector Machines for Large-Scale Regression Problems", *J. Machine Learning Research*, Vol. 1, 2001, pp. 143-160.

[11] A.P. Dempster, N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *J. Royal Statistical Soc., Ser. B*, Vol. 39, No. 1, 1977, pp. 1-38.

[12] G. R. Doddington, M. A. Przybycki, A. F. Martin and D. A. Reynolds, "The NIST speaker recognition evaluation - Overview, methodology, systems, results, perspective", *Speech Commun.*,Vol. 31, No. 2-3, 2000, pp. 225-254.

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

[14] J.-L. Dugelay, J.-C. Junqua, C. Kotropoulos, R. Kuhn, F. Perronnin and I. Pitas, "Recent Advances in Biometric Person Authentication", *Proc. ICASSP*, Orlando, 2002, pp. 4060-4062 (Vol. IV).

[15] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *Proc. Second European Conference on Computational Learning Theory*, 1995.

[16] M. Gales, "The generation and use of regression class trees for MLLR adaptation", TR 263, Cambridge Univ. Engin. Dept., 1996.

[17] J. L. Gauvain and C.-H. Lee, "Maximum A Posteriori estimation for multivariate Gaussian mixture observation of Markov chains", *IEEE Trans. Speech and Audio Processing*, Vol. 2, No. 2, 1994, pp. 291-298.

[18] T. Joachims, "Making Large-Scale SVM Learning Practical" in: *Advances in Kernel Methods - Support Vector Learning* (editors: B. Schölkopf, C. Burges and A. Smola), MIT-Press, 1999.

[19] J. Kittler, M. Hatef, R.P.W. Duin and J. Matas, "On Combining Classifiers", *IEEE Trans. Pattern Analysis and Machine Intell.*, Vol. 20, No. 3, 1998, pp. 226-239.

[20] R. Kuhn, P. Nguyen, J.C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field and M. Contolini, "Eigenvoices for Speaker Adaptation", *Proc. ICSLP*, 1998, pp. 1771-1774.

[21] M. Lockie (editor), "Facial verification bureau launched by police IT group", *Biometric Technology Today*, Vol. 10, No. 3, 2002, pp. 3-4.

[22] J. Lüttin and G. Maître, "Evaluation protocol for the extended M2VTS database (XM2VTSDB)", *IDIAP-Com 98-05*, 1998.

[23] J. Mariéthoz and S. Bengio, "A Comparative Study of Adaptation Methods for Speaker Verification", *Proc. ICSLP*, Denver, 2002, pp. 581-584.

[24] J. Mariéthoz, D. Genoud, F. Bimbot and C. Mokbel, "Client / World Model Synchronous Alignement for Speaker Verification", *Proc. EUROSPEECH*, Budapest, 1999, pp. 1979-1982 (Vol. 5).

[25] D. Reynolds, T. Quatieri and R. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing*, Vol. 10, No. 1-3, 2000, pp. 19-41.

[26] C. Sanderson and S. Bengio, "Robust Features for Frontal Face Authentication in Difficult Image Conditions", *IDIAP-RR 03-05*, 2003.

[27] C. Sanderson and K. K. Paliwal, "Information Fusion and Person Verification Using Speech and Face Information", *IDIAP-RR 02-33*, 2002.

[28] C. Sanderson and K.K. Paliwal, "Polynomial Features for Robust Face Authentication", *Proc. ICIP*, Rochester, 2002, pp. 997-1000 (Vol. 3).

[29] M. Sharma and R. Mammone, "Subword-Based Text-Dependent Speaker Verification System With User-Selectable Passwords", *Proc. ICASSP*, Atlanta, 1996, pp. 93-96 (Vol. 1).

[30] M. Turk and A. Pentland, "Eigenfaces for Recognition", *J. Cognitive Neuroscience*,Vol. 3, No. 1, 1991, pp. 71-86.

[31] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, 1999 (2nd ed.).

[32] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. Information Theory*, Vol. 13, No. 2, 1967, pp. 260-269.

[4] Torch is available under a BSD license from www.torch.ch

** IDIAP Research Reports (RR) are available via www.idiap.ch