Fast Speaker Verification on Mobile Phone data using Boosted Slice Classifiers

Anindya Roy[1,2], Mathew Magimai.-Doss[1], Sébastien Marcel[1]
[1]Idiap Research Institute, Martigny, Switzerland
[2]École Polytechnique Fédérale de Lausanne, Switzerland
Anindya.Roy@idiap.ch, mathew@idiap.ch, Sebastien.Marcel@idiap.ch

Abstract

In this work, we investigate a novel computationally efficient speaker verification (SV) system involving boosted ensembles of simple threshold-based classifiers. The system is based on a novel set of features called "slice features". Both the system and the features were inspired by the recent success of pixel comparison-based ensemble approaches in the computer vision domain. The performance of the proposed system was evaluated through speaker verification experiments on the MOBIO corpus containing mobile phone speech, according to a challenging protocol. The system was found to perform reasonably well, compared to multiple state-of-the-art SV systems, with the benefit of significantly lower computational complexity. Its dual characteristics of good performance and computational efficiency could be important factors in the context of SV system implementation on portable devices like mobile phones.

1. Introduction

Today, speaker verification (SV) systems are finding their way into mobile phones and other portable devices [5][6]. This has led to the following objectives: a) to ensure robustness of the system against additive noise (because such devices are liable to be used anywhere, even in very noisy acoustic environments) as well as channel and session variabilities, and b) to keep the computations light enough to be implementable on such devices. To fulfill the first objective, ie. robustness, the basic SV framework [1], involving cepstral features modelled by a Gaussian Mixture Model (GMM) - Universal Background Model (UBM) framework, is often reinforced by methods such as feature warping [3], Support Vector Machine (SVM) with supervector kernels, Joint Factor Analysis (JFA) [4] and score normalization [1]. Though such enhancements make the system more robust, they may pose a problem for the second objective, i.e. computational efficiency.

In this work, we investigate a novel SV system which addresses both objectives of robustness and computational efficiency at the same time. The inspiration for this work is the recent success of ensemble learning methods involving features based on pixel comparison like Haar features, Local Binary Patterns (LBP) [8] and Fern features [7] in the computer vision community. In this work, we study a similar set of features called "slice feature" which calculates the difference in magnitude at two frequency points in the speech spectrum. Simple threshold-based classifiers trained on such slice features are iteratively selected by the Adaboost algorithm [2] giving a final classifier which is a simple weighted sum of comparison operations. We call this the Boosted Slice Classifier (BSC) system.

This system was originally proposed by the authors in a previous work [9] which showed that it performed well compared to baseline MFCC-GMM systems for an SV task using the XM2VTS database [9]. In the current work, we extend this study by performing text-independent SV experiments on the more recent and challenging MOBIO database containing speech collected with mobile phones [6]. This database was used in the MOBIO Face and Speaker Verification Evaluation at ICPR 2010.[1] Compared to the previous study, a much more difficult protocol was followed, involving mismatch at multiple levels [6]. Also, unlike the MFCC-GMM baseline system in [9], the BSC system is compared in the current work with state-of-the-art SV systems. Finally, the computational complexity of the proposed and state-of-the-art systems is analysed. It is found that the BSC system shows comparable SV performance at significantly lower computational complexity, compared to the state-of-the-art systems.

The rest of the paper is organized as follows. In Section 2, we describe the BSC framework. We describe our experiments in Section 3. In Section 4, we analyse and compare the computational complexity of our method. Finally, Section 5 discusses the main conclusions of our work.

[1]www.mobioproject.org/icpr-2010

2. The BSC Framework

The framework was originally described in our previous work [9]. Since it is relatively new, we describe it again for convenience. In the current description, we refine the concept of "binary feature" [9] in terms of "slice" and "slice classifier".

2.1. Feature representation: The concept of slice

Firstly, the input speech waveform is blocked into frames and windowed. Silence frames are discarded. Fourier transform is applied, yielding a sequence of spectral magnitude vectors. Let $\mathbf{X} = [X(1), \cdots, X(N_{\mathrm{X}})]^T$ be an instance of such a spectral vector, where N_{X} denotes the size of the vector. In particular, let $\mathbf{X}_j$ denote the j-th vector in the sequence. Given the spectral vector $\mathbf{X}$, slice feature L_i is calculated from $\mathbf{X}$ as follows:

$$L_i \equiv L_i(\mathbf{X}) = \mathrm{X}(k_{i,1}) - \mathrm{X}(k_{i,2}). \tag{1}$$

where $\{k_{i,1}, k_{i,2}\}$ is an ordered pair of frequency points uniquely associated with slice feature L_i. The parameters $k_{i,1}$, $k_{i,2}$ can vary from 1 to N_{X} but cannot be equal. This constraint restricts the total number of slice features as defined above to $N_L = N_{\mathrm{X}}(N_{\mathrm{X}} - 1)$. Let $L_i(\mathbf{X})$ be denoted by L_i and $L_i(\mathbf{X}_j)$ be denoted by $L_{i,j}$.

2.2. Feature modelling: Slice Classifiers

Each slice L_i has an associated slice classifier f_i. The classifier is a simple hard threshold classifier with a single parameter, the threshold θ_i. This classifier can 'see' instances of only slice L_i and it has to classify these as either belonging to the client ('1') or an impostor ('0'). The output of f_i is calculated as,

$$f_i(L_i) = \begin{cases} 1 \text{ (client)} & \text{if } L_i \geq \theta_i, \\ 0 \text{ (impostor)} & \text{otherwise.} \end{cases} \tag{2}$$

Training classifier f_i involves selecting threshold θ_i that will minimize misclassification error ϵ_i on a given training set of slice values extracted from client and impostor spectral vectors. The optimal θ_i value can be found in a single pass by a search over the sorted slice values. Note that total number of slice classifiers is same as total number of slices, N_L.

2.3. Classifier Selection by Discrete Adaboost

Out of all the slice classifiers, a small number of slice classifiers $N_L^* \ll N_L$ are iteratively selected *for each client* according to their discriminative ability with respect to that client. This selection is based on the Discrete Adaboost algorithm [2] with weighted resampling, which is widely used for such binary feature selection tasks [8] and is known for its robust performance [2]. The algorithm, which is to be run once for each client, is detailed as follows:

Algorithm: Slice classifier selection by Discrete Adaboost

Inputs: 1) N_{tr} training samples (spectral vectors) $\{\mathbf{X}_j\}_{j=1}^{N_{tr}}$, 2) the corresponding class labels, $y_j \in \{0,1\}$ (0:*impostor*, 1:*client*), 3) N_L^*, the number of slice classifiers to be selected, 4) N_{tr}^*, the number of training vectors to be randomly selected at each iteration ($N_{tr}^* \ll N_{tr}$) [2].
Steps:
1. Initialize the training sample weights:
$\{w_{1,j}\} \leftarrow \frac{1}{2N_{tr}^{(0)}}, \frac{1}{2N_{tr}^{(1)}}$ for $y_j = 0, 1$ respectively and $1 \leq j \leq N_{tr}$. $N_{tr}^{(0)}$ and $N_{tr}^{(1)}$ are the number of impostor and client training vectors respectively.
2. Repeat for $n = 1, 2, \cdots N_L^*$:

- Normalize sample weights, $w_{n,j} \leftarrow \frac{w_{n,j}}{\sum_{j'=1}^{N_{tr}} w_{n,j'}}$
- Randomly select a subset of N_{tr}^* training samples, according to the probability distribution given by weights $\{w_{n,j}\}$
- From this subset, extract slice features, $\{L_{i,j}\}_{i=1,j=1}^{N_L, N_{tr}^*}$ as in Equation 1. For each slice L_i, train a threshold classifier f_i. Let misclassification error of each f_i be ϵ_i.
- Select the next best slice classifier, $f_n^* = f_{i^*}$ and its associated slice $L_n^* = L_{i^*}$ where $i^* = \arg\min_i \epsilon_i$. In other words, select the next best slice classifier as that one which has currently the lowest misclassification error ϵ_i.
- Set $\beta_n \leftarrow \frac{\epsilon_{i^*}}{1-\epsilon_{i^*}}$ where ϵ_{i^*} is the misclassification error of the selected slice classifier.
- Update *all* training sample weights,
$w_{n+1,j} \leftarrow w_{n,j} \beta_n^{\mathbf{1}_{\{f_n^*(L_{n,j}^*) = y_j\}}}$ for $1 \leq j \leq N_{tr}$.
- Set the selected slice classifier weight,
$\alpha_n = -\log(\beta_n)$.

3. Normalize slice classifier weights,
$\alpha_n \leftarrow \frac{\alpha_n}{\sum_{n'=1}^{N_L^*} \alpha_{n'}}$, for $1 \leq n \leq N_L^*$.
Outputs: 1)The sequence of best slice classifiers $\{f_n^*\}_{n=1}^{N_L^*}$ selected by the algorithm, along with their thresholds $\{\theta_n\}_{n=1}^{N_L^*}$, 2) Their associated slices, $\{L_n^*\}_{n=1}^{N_L^*}$ defined by their parameters $\{k_{n,1}, k_{n,2}\}_{n=1}^{N_L^*}$. 3) Classifier weights $\{\alpha_n\}_{n=1}^{N_L^*}$.

---

[2]A value of N_{tr}^* equal to 5% of N_{tr} was found to work well for all experiments reported here in subsequent sections.

2.4. Slice classifier combination

For each client, the selected slice classifiers are combined linearly to form a strong classifier, F [2]. Let $\mathbf{X}$ be a test spectral vector extracted from an utterance, U. Then the strong classifier score is calculated as a linear sum,

$$F = \sum_{n=1}^{N_L^*} \alpha_n f_n^*(L_n^*(\mathbf{X})). \tag{3}$$

Scores from each frame in the utterance are added and normalized by number of frames N_{fr}, to obtain the final score for the utterance. This is compared with a preset threshold Θ to decide if the utterance was made by a client or an impostor. This threshold Θ is set based on the Equal Error Rate (EER) [1] on a development set of speakers that is distinct from the test set (ref. Section 3.3).

3. Experiments

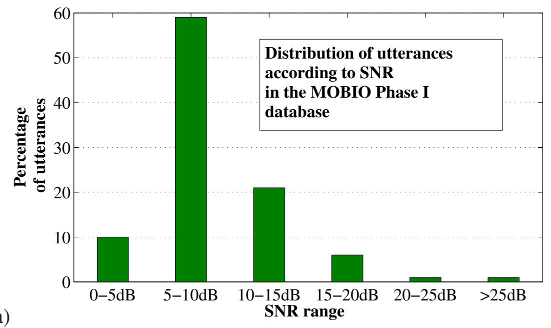
3.1. Database description

Experiments were performed on the MOBIO Phase I database [6][5] which consists of speech data collected from 152 people (100 males, 52 females) using mobile phones. The data was collected at 6 different sites in 5 different countries. There were both native and non-native English speakers. The sampling frequency was 48 kHz.[3] Data for each speaker was collected in 6 separate sessions, with a gap of at least one month between sessions. In each session, the speakers were asked to answer a set of 21 questions. There were 3 types of questions: a) 5 questions requiring 5 short set response answers (read speech from the mobile display), b) 1 question requiring 1 long set response answer (read speech from a paper), and c) 15 questions each requiring free speech answer. Each answer was recorded as one utterance.

This database has the following challenges.

- All speech data was collected on mobile phones and had significant amount of noise [6]. About 10 % of the utterances had SNRs less than 5 dB, while 60 % had SNRs between 5 to 10 dB (see Figure 1(a)).
- Utterances had limited amount of speech. About 25 % of utterances had less than 2 seconds of speech, while 35 % had between 2 to 3 seconds of speech (see Figure 1(b)).
- The data presented possibilities for testing different levels of mismatch using a challenging protocol. This is further explained in Section 3.3.

[3]However, it was downsampled to 8 kHz for all experiments reported here for the proposed BSC system.



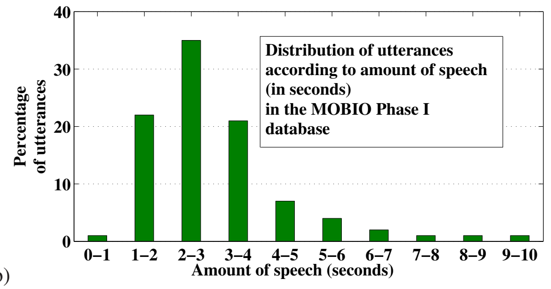


Figure 1. Distribution of utterances in the MOBIO Phase I database, according to (a) their SNR (dB), (b) amount of speech in seconds.

System	Feature dimension, N_D	No. of Gaussians in the GMM, N_G
BUT 1, BUT 2	60	2048
LIA 1, LIA 1a	70	512
LIA 2, LIA 2a	50	256
TEC-ASU 1	33	512
TEC-ASU 2	49	512
UWB 1, UWB 2	40	510
SUV 1, SUV 1a	59	512
SUV 2	33	512

Table 1. Basic parameters of the reference systems, grouped according to submitting institution. Please see Section 3.2 for details.

3.2. Description of systems tested

The proposed BSC system was compared with 17 state-of-the-art reference systems from 5 independent research groups: 1) Brno University of Technology (BUT), 2) The University of Avignon (LIA), 3) Tecnologico de Monterrey, Mexico and Arizona State University, USA (TEC-ASU), 4) The University of West Bohemia (UWB), and 5) Swansea University and Validsoft (SUV).[4] All of these participated in the MOBIO evaluation at ICPR 2010. All their system details are provided in [6][5].

[4]Henceforth, reference systems shall be denoted by the format "group-name system-number", for example, BUT 1, BUT 2, LIA 3, etc.

Training set		
Session Number	Usage	Data to use
1-6	Background training	All data

Development & Test set		
Session Number	Usage	Data to use
1	Enrolment	Set questions only
2-6	Generate test scores	Free speech only

Table 2. Usage of data in Training, Development and Test splits of the MOBIO database [6]. Please see Section 3.3 for details.

Here, we highlight the chief aspects of these reference systems for convenience. All reference systems used cepstral features [1]. Systems varied in the number of filterbanks (ranging from 24 to 50), the number of cepstral features (16 to 29) and the use of delta and delta-delta cepstra. The final feature dimension varied from 33 to 70 (ref. Table 1). Systems also varied in the kind of feature normalization and feature warping used. All rererence systems (except one) used GMM-UBM as the primary modelling block. Number of Gaussians in the GMM varied from 256 to 2048 (ref. Table 1). System UWB 3 used 3rd-order polynomial expansion resulting in a 12341-dimensional supervector. A majority of reference systems (BUT 1,2,3, LIA 1,1a,2,2a, UWB 2,3,4) used secondary modelling blocks like supervector SVM with Joint Factor Analysis, or iXTractor system. Most systems also used some kind of score normalization like s-norm, z-norm or t-norm. Some systems like BUT 3, UWB 4, SUV 3 were fusions of other systems submitted by the same group.

For the proposed BSC system, a 256-point Fourier transform was applied to each speech frame. One half of the symmetric magnitude spectrum was retained, yielding spectral vectors $\mathbf{X}$ of length $N_\mathrm{X} = 128$. Thus, the total number of slice classifiers, $N_L = N_\mathrm{X}(N_\mathrm{X} - 1) = 16256$ (ref. Section 2.1). Out of this, the number of slice classifiers N_L^* selected by Adaboost for each client was approximately 100. Increasing the number of selected slice classifiers beyond this value did not result in further improvement of performance.

3.3. Experimental Protocol

The SV protocol used was the same as in the MOBIO Face and Speaker Verification Evaluation, details of which are given in [5][6]. Here, we highlight the chief aspects of this protocol. The database is split into three distinct sets: training set, development set and test set. The 3 sets are completely separate in terms of speakers and data collection sites. The purpose of each set is described below.

The purpose of training set was to derive background models or JFA subspaces for reference systems and for providing negative ('0') samples while boosting each client model for the BSC system. Purpose of development set was to derive an EER-based threshold while purpose of test set was to evaluate the system performance using this threshold.

The development and test sets had their own distinct set of clients. The protocol for enroling and testing were the same for both sets. Only 5 set response questions from session 1 could be used to enrol a client. Thus, they provided the positive ('1') samples while boosting a client model for BSC system. Testing was then conducted on all 15 *free* speech questions from sessions 2 to 6 each, equalling 75 test utterances per client. When producing imposter scores all the other clients were used as imposters. The performance was calculated in terms of the Half Total Error Rate (HTER) on the test set. Separate experiments for male and female speakers were conducted. In Table 2, a brief summary of the usage of data for the training, development and test sets is provided.

The protocol for MOBIO presents some special challenges in addition to the noisy data itself. They are as follows.

- *Session variability.* Only a single session per client could be used to train (enrol) the target speaker models. Testing was done on remaining five sessions.
- *Lexical mismatch* Speech used in enrolment and testing had different lexical content (*text-independent* SV problem).
- *Speech-type mismatch.* The training (enrolment) was done on read speech while testing was on free speech.
- *Site mismatch.* All background (impostor) data allowed for *training* came from 2 sites while all impostor data used for *testing* came from the 4 remaining sites.

3.4. Results

The Half Total Error Rate (HTER %) on the test set of the MOBIO database for all 18 systems have been shown in Figure 2. In all cases, performance of the proposed BSC system is reasonably good, close to the mean of reference systems' performance.

It is noteworthy that the proposed system achieved this using a very simple framework whereas all the state-of-the-art reference systems used more sophisticated techniques involving more computations (discussed in Section 4). This indicates that the BSC system achieves a good trade-off between system performance and computational efficiency.

4. Discussion on Computational Complexity

In this section, we compare the computational complexity of the proposed BSC system with that of the reference

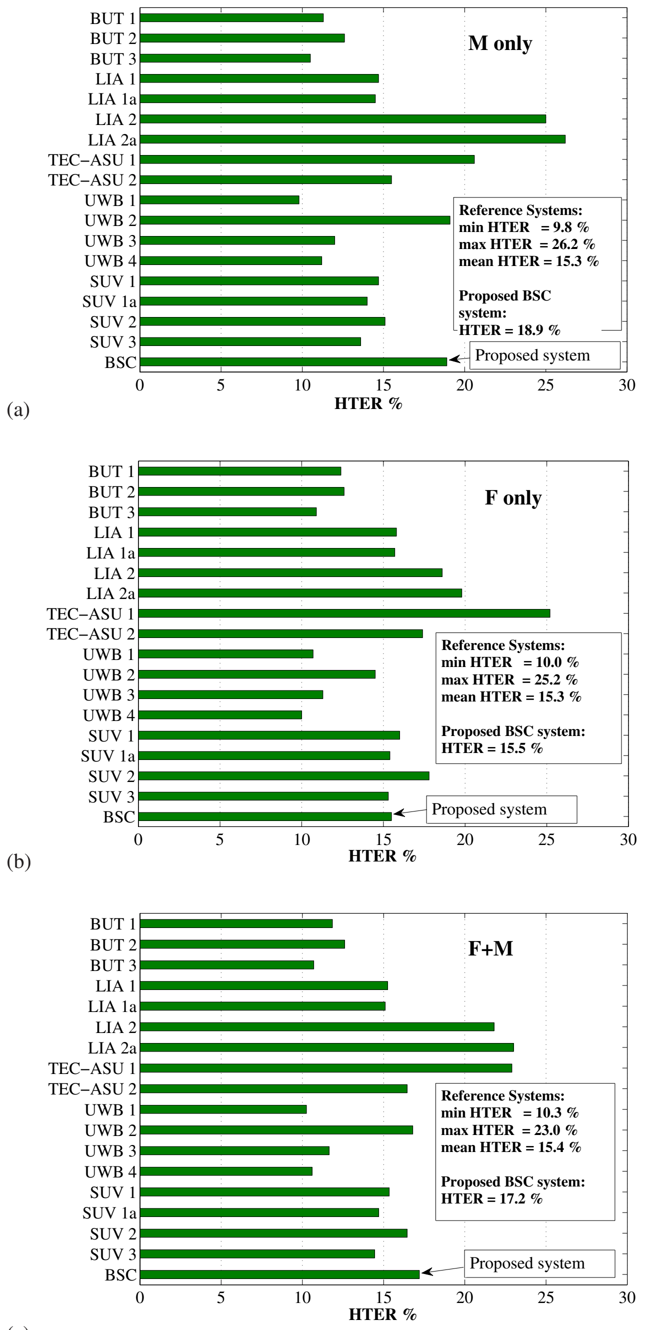


Figure 2. Half Total Error Rates (HTER %) for SV experiments on the Test set of the MOBIO Phase I database, using (a) only male speakers, (b) only female speakers, and (c) average of the two. HTERs are shown for the 17 reference systems and proposed BSC system. Please consult the text (Section 3) for more details.

systems mentioned in Section 3. We consider only the client access (or *test*) phase because it is online, as opposed to the training phase which is offline. For this, we count the number of floating-point operations (FLOP) starting from *after* the feature extraction stage till the calculation of the final score at a frame level. In fact BSC system has a simpler feature extraction stage, with no filterbanks nor feature warping. For simplicity, we ignore this.

For reference systems, we consider only the essential modelling block while computing number of FLOPs, i.e. only the computation of the Gaussian components for GMM-based system. We ignore all other blocks, such as those related to factor analysis, iXTractor, supervector SVM, etc. which are present in a majority of reference systems. We do this for keeping the analysis simple, at the cost of a pessimistic bias against our system.

For reference GMM systems, it can be shown that the number of floating point multiplications $n^{\times}$ and additions n^{+} involved in processing one frame is: $n^{\times} = 2N_G N_D$, $n^{+} = N_G(2N_D - 1)$. where N_D is feature dimension and N_G is number of Gaussians. Hence, the total number of FLOPs is:

$$N_{\text{FLOP}} = n^{\times} + n^{+} = N_G(4N_D - 1). \qquad (4)$$

In addition to multiplications and additions, a small number of exponentiations may also be required (via `log-add` operation) but we ignore this, because the precise number of exponentiations is not fixed.

For the BSC system, let $\mathbf{X}$ be the spectral vector extracted from a frame and N_L^* be the number of slice classifiers selected. Then the frame-level score F is computed as follows (ref. Equations 1, 2 and 3):

```
$F \leftarrow 0$
for $n = 1$ to $N_L^*$
    $a \leftarrow \{0\ ,\ \alpha_n\}$
    $b \leftarrow (X(k_{n,1}) - X(k_{n,2}) \geq \theta_n)$
    $F \leftarrow F + a[b]$
end
```

Here, $a[b]$ denotes the b-th element of array a. Since they usually take almost the same time, we group the number of comparisons, additions and subtractions into n^{+}. From the above implementation, we find that for the BSC system, no multiplication is required and,

$$N_{\text{FLOP}} = n^{+} = 3N_L^* \qquad (5)$$

The total number of FLOPs for BSC and reference systems calculated using Equations 4 & 5 are shown in Figure 3. Parameter values for N_D, N_G in Equation 4 are enlisted in Table 1. In Equation 5, parameter $N_L^* = 100$ (ref. Section 3.2).

It is observed from Figure 3 that BSC system requires a few hundred FLOPs, significantly less than that required by reference systems ($10^4 - 10^5$ FLOPs). Hence, even with a

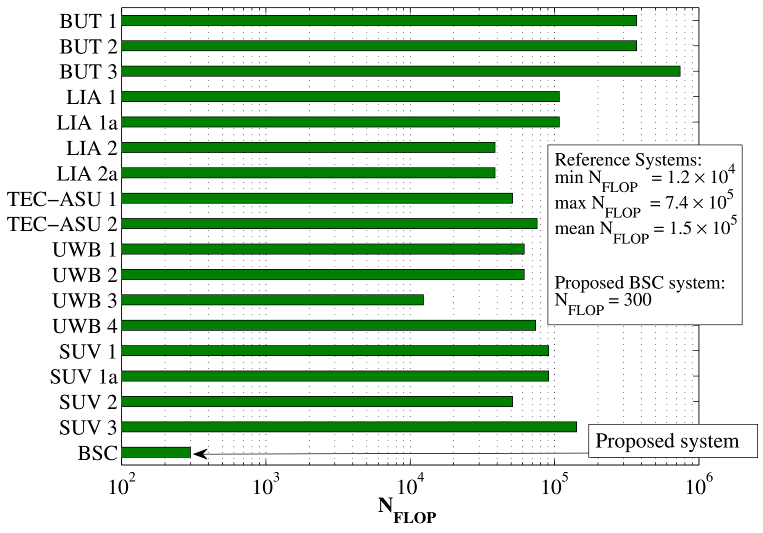


Figure 3. Number of floating-point operations, N_{FLOP} for the 17 reference systems and the proposed BSC system, plotted in log-scale.

pessimistic bias, BSC system is shown to be computationally more efficient. This is an important advantage of the BSC system particularly with respect to the computational constraints for mobile phone SV systems (Section 1).

5. Conclusions

We investigated the performance of the boosted ensemble-based BSC system on a challenging speaker verification task using the MOBIO mobile phone speech database and compared it with several state-of-the-art MFCC-GMM based systems. The BSC system showed comparable performance, but involved significantly less number of computations.

Hence, it seems to fulfill the two objectives related to implementation of SV systems on portable devices such as mobile phones, ie. robustness and computational efficiency. Further work will aim at improving the SV performance of the proposed system by investigating different feature representations and classifier selection and combination approaches.

6. Acknowledgements

The authors would like to thank the Swiss National Science Foundation, projects MultiModal Interaction and MultiMedia Data Mining (MULTI, 200020-122062) and Interactive Multimodal Information Management (IM2, 51NF40-111401) and the FP7 European MOBIO project (IST-214324) for their financial support.

References

[1] F. Bimbot et al. A Tutorial on Text-Independent Speaker Verification. *EURASIP Journal on Applied Signal Processing*, 4:431–451, 2004. 1, 3, 4

[2] J. Friedman, T. Hastie, and R. Tibshirani. Additive Logistic Regression: a Statistical View of Boosting. *Annals of Statistics*, 28:2000, 1998. 1, 2, 3

[3] J. Pelcanos and S. Sridharan. Feature warping for robust speaker verification. In *2001: A Speaker Odyssey Workshop*, June 2001. 1

[4] P. Kenny et al. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio Speech and Language Processing*, 15(4):1435, 2007. 1

[5] S. Marcel, C. McCool, P. Matejka, T. Ahonen, and J. Cernocky. Mobile biometry (MOBIO) face and speaker verification evaluation. Idiap Research Report Idiap-RR-09-2010, Idiap, May 2010. 1, 3, 4

[6] S. Marcel, C. McCool, P. Matejka, T. Ahonen, and J. Cernocký. On the results of the first mobile biometry (mobio) face and speaker verification evaluation. In *Proc. of ICPR*, pages 210–225, 2010. 1, 3, 4

[7] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast Keypoint Recognition using Random Ferns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(3):448–461, 2010. 1

[8] Y. Rodriguez. Face Detection and Verification using Local Binary Patterns. PhD Thesis 3681, Ecole Polytechnique Federale de Lausanne, 2006. 1, 2

[9] A. Roy, M. Magimai-Doss, and S. Marcel. Boosted binary features for noise-robust speaker verification. In *Proc. of IEEE ICASSP*, pages 4442–4445, 2010. 1, 2