A Fast Parts-based Approach to Speaker Verification using Boosted Slice Classifiers

Anindya Roy*, *Student Member, IEEE,* Mathew Magimai.-Doss, *Member, IEEE,*
and Sébastien Marcel, *Member, IEEE*

***Abstract*—Speaker verification on portable devices like smartphones is gradually becoming popular. In this context, two issues need to be considered: 1) such devices have relatively limited computation resources, and 2) they are liable to be used everywhere, possibly in very noisy, uncontrolled environments. This work aims to address both these issues by proposing a computationally efficient yet robust speaker verification system. This novel parts-based system draws inspiration from face and object detection systems in the computer vision domain. The system involves boosted ensembles of simple threshold-based classifiers. It uses a novel set of features extracted from speech spectra, called "slice features". The performance of the proposed system was evaluated through extensive studies involving a wide range of experimental conditions using the TIMIT, HTIMIT and MOBIO corpus, against standard cepstral features and Gaussian Mixture Model-based speaker verification systems.**

***Index Terms*—EDICS: BIO-EXPR (Human Identification based on voice or handwriting), Speaker verification, binary features, speaker-specific features, parts-based approach, noise robustness, Adaboost, feature selection, mobile biometrics, computational efficiency**

I. Introduction

Today, speaker verification (SV) systems are gradually becoming more and more ubiquitous, finding their way into smartphones and other portable devices [1] [2] [3]. This has led to the following objectives: a) robustness against a noisy acoustic environment (additive noise) as well as channel and session variabilities, and b) computational efficiency (i.e. the computations must be light enough to be implementable on such devices).

To fulfill the first objective, ie. robustness, the basic Gaussian Mixture Model (GMM) - Universal Background Model (UBM) SV framework [4] is often augmented by feature normalization [5], model normalization [6] [7] and score normalization [8]. However, improved robustness of such systems comes at the cost of more computations which may pose a task with respect to the second objective, ie. computational efficiency. Hence, the question is how to fulfill both the objectives at the same time.

A possible answer comes from a class of object detection algorithms developed in the computer vision domain in recent years. These algorithms use simple localized features based on comparison of image *parts*. These algorithms have the following characteristics relevant to our objectives: a) robustness in difficult test scenarios, involving uncontrolled illumination variations, and b) computational efficiency (they are faster compared to older approaches such as Eigenfaces [9], Fisherfaces [10], etc.). Three representative algorithms from this class are as follows: a) Rapid Object Detection using a boosted cascade of Haar features [11] [12], b) Fast Key-point Recognition using Random Fern features [13], and c) Face Detection and Verification using Local Binary Patterns [12].

In this work, we have drawn inspiration from all these algorithms. For clarity, we provide a brief description of the relevant aspects of these algorithms. They combine decisions from a set of classifiers, each of which look at specific parts of the entire object (or feature space). Each classifier involves the comparison of intensities in two parts of the object. These could be the intensities at two pixel locations (Fern features) or the average intensities over two patches of pixel locations (Haar features). The decisions from these classifiers are binary. Suitable ensemble learning approaches such as Adaboost [14] are often used to select the classifiers which are most discriminative with respect to the task (face or object detection).

In this work, this framework is ported to the SV domain in the following way. The 1-D spectral vectors derived from speech are equivalent of the 2-D images. As in the vision domain, the algorithm combines decisions from a set of classifiers. Instead of looking at parts of an image, here the classifiers look at parts of spectral vectors, precisely the spectral magnitudes at pairs of frequency points. The Discrete Adaboost algorithm is used to select the most discriminative classifiers. This is the central idea of our approach.

This approach was originally proposed by the authors in a previous work [15] which showed that it performed well compared to baseline GMM-UBM systems for an SV task using XM2VTS database. Since it is a relatively new approach in the SV domain, this work provides a detailed description of this approach, refining the original concept of "binary feature" [15] in terms of "slice" and "slice classifier". In addition, this work extends the previous study in three directions.

Firstly, more extensive experiments were carried out to evaluate the text-independent SV performance of the system on clean speech and speech corrupted by additive noise and channel noise (using the TIMIT [16], HTIMIT [17] and noisy TIMIT databases) as well as speech data collected using mobile phones (using the MOBIO database [1] [2]

A. Roy is with the Doctoral School of Electrical Engineering, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, and Idiap Research Institute, Martigny, Switzerland. E-mail: anindya@ieee.org

M. Magimai.-Doss and S. Marcel are with Idiap Research Institute, Martigny, Switzerland.

[3]). Secondly, the proposed approach was compared against both baseline GMM-UBM system as well as state-of-the-art SV systems [2]. Thirdly, we carried out detailed analyses of the proposed approach in terms of a) robustness in noisy scenarios, b) computational complexity, and c) the distribution of discriminative parts selected by the algorithm. The proposed approach has performed favorably with all the experimental conditions and databases tested, and compares well with baseline and state-of-the-art SV systems.

The rest of the paper is organized as follows. A brief overview of standard SV systems and the proposed system is given in Sections II and III respectively. The proposed system is described in detail in Section IV. Section V gives a brief overview of the experiments carried out, while Sections VI and VII describe these experiments in detail. Section VIII analyses some aspects of the proposed system while Section IX concludes the work.

II. BRIEF OVERVIEW OF STANDARD SV SYSTEMS

Standard SV systems use Mel Frequency Cepstral Coefficients (MFCC) or Linear Prediction Cepstral Coefficients (LPCC) [18] as their features. These features characterize the shape of the short-time log spectrum of speech. This is done by processing the estimated log spectrum through an energy-compacting and decorrelating transform like the Discrete Cosine Transform (DCT) or the Fast Fourier Transform (FFT) and retaining only the first few highest energy coefficients (typically 13 to 19).[1] Because of this transform, every region of the spectrum contributes to *each* cepstral feature. Hence, such features could be termed holistic and noise in one *part* of the spectrum could affect *all* the cepstral features.

These cepstral features are modelled by GMM-UBM [4]. Typically, speech samples from a large set of speakers (called the “world” set) distinct from the client is used to train a GMM. This is called the Universal Background Model (UBM). Next, for each client, client speech samples are used to adapt (typically only the means of) this UBM, to create the client-specific GMM. During test, log likelihood ratio of the test samples using the UBM and the client-specific GMM is compared with a pre-set threshold. Based on this comparison, the speaker is classified as the true client or an impostor. This modeling of cepstral features using GMM forms the basic framework of standard SV systems. In this work, we call this the “baseline system”.

To achieve robustness against acoustic noise and channel and session variabilities, this baseline system is often augmented a) at the feature level by feature warping [19], b) at the model level by meta-modelling approaches such as Support Vector Machines with GMM Supervector (GSV) kernel [6] or Generalized Linear Discriminant Sequence (GLDS) kernel [20], Latent Factor Analysis (LFA) [21], Joint Factor Analysis (JFA) [7] and I-vector system [22], and c) at the decision-making level by score normalization techniques [8] such as Z-norm and T-norm. These augmented systems are representative of the current state-of-the-art [1] and hence are called “state-of-the-art system” in this work. Figure 1 (a) provides a block diagram of the baseline and state-of-the-art SV systems.

III. BRIEF OVERVIEW OF PROPOSED SV SYSTEM

The proposed SV system consists of four stages: 1) feature representation using slice features, 2) feature modelling by slice classifiers, 3) slice classifier selection and 4) final classifier. We briefly describe each of these below and compare them with standard SV system. For clarity, some basic notations related to each stage are also introduced.

A. Feature representation using slice features

The starting point of our system is the short-time spectrum extracted from speech. The difference in magnitude at two distinct frequency points in the short-time spectrum is taken as a feature, termed as “slice” feature. Enumerating all possible pairs of distinct frequency points in the spectrum generates the complete list of such slice features.[2]

A slice feature is denoted as L_i where i is an index to the complete list of slice features. Each slice L_i must be uniquely associated with an ordered pair of frequency points, denoted by $\{k_{i,1},\ k_{i,2}\}$.

It is noteworthy that each feature looks at only certain parts of the spectrum (precisely, *two* frequency points). Hence, it is localized or *parts-based*. It is unaffected if *other* parts of the spectrum are affected by noise. This behaviour contrasts with the holistic cepstral features in standard SV systems (ref. Sec. II).[3]

B. Feature modelling by slice classifiers

Each slice feature is discriminatively modelled for each client speaker. More precisely, for each slice feature, a simple threshold-based classifier (termed a “slice classifier”) is trained to classify as true client (‘1’) or impostor (‘0’), based on *only* that slice. The two-class (‘0’-‘1’) decisions of these slice classifiers are termed as “binary features”. Let f_i be the slice classifier associated with the slice L_i.

The conceptual relation between features and classifiers is depicted as follows:

$$\begin{array}{ccccccc} \text{Slice} & & \text{Client-} & & \text{Slice} & & \text{Binary} \\ \text{Feature,} & + & \text{specific} & \rightarrow & \text{Classifier,} & \rightarrow & \text{Feature} \\ L_i & & \text{threshold} & & f_i & & \text{(‘0’-‘1’)} \end{array}$$

Further stages in the system are unaffected as long as the noise in the speech signal is not so high as to change the decision of these classifiers, ie. the binary features. On the other hand, cepstral features in standard SV systems could be affected even when there is a small amount of noise (ref. Sec. VIII-A).[4]

[1] In the case of LPCC, a more direct method is used. However, it is mathematically equivalent to taking the FFT of the estimated log spectrum.

[2] There is no restriction on the frequency point locations. Hence, the total number of slice features depends only on the number of the frequency points in the spectrum.

[3] This contrast is analysed in more detail in Section VIII-A.

[4] This is analogous to the comparison between analogue and digital systems.

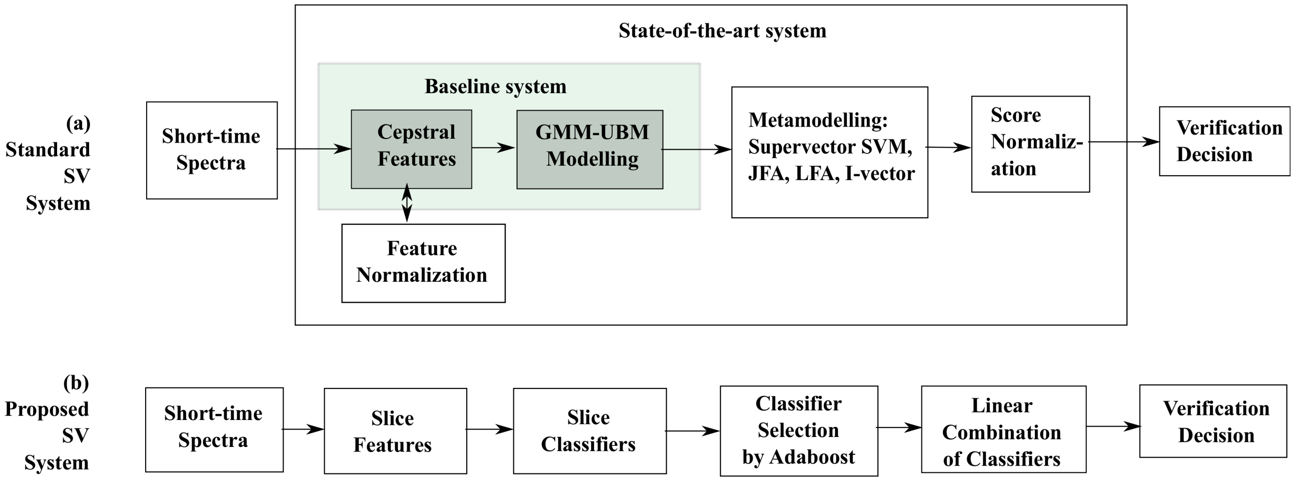


Fig. 1. Block diagrams showing the main stages of (a) standard SV systems and (b) proposed SV system. In the case of standard SV systems (a), the baseline (basic) system and the state-of-the-art systems are depicted. Please consult the text (Sections II, III) for details.

C. Classifier selection

The total number of slice features (and hence slice classifiers) are typically quite high (16256 for a 256-point FFT spectrum). Most of these features may not contain any discriminative information pertaining to the client speaker. In other words, the corresponding slice classifiers may perform poorly. But a few will contain such discriminative information and their classifiers will perform relatively better.

The Discrete Adaboost algorithm is used to iteratively select such discriminative slice classifiers, based on their performance on increasingly misclassified training samples. This stage also has no direct counterpart in standard SV systems.

D. Final classifier

The final classifier denoted by F is a simple weighted sum of the '0'-'1' outputs of individual slice classifiers which were selected in the previous stage. During test, the final classifier output is compared with a pre-set threshold to classify the speaker as client or impostor. We note that this classifier is much simpler compared to classifiers in standard SV systems (ref. Sec. II). This comparison is analysed in detail in Section VIII-B.

Figure 1 (b) provides a block diagram of the proposed system. We call the system as the Boosted Slice Classifier (BSC) system because it involves selecting (boosting) slice classifiers. As described above, some of the stages in the proposed system (feature modelling and classifier selection) do not exist in standard SV systems, while the rest are different from their counterparts in standard SV systems. This shows the originality and novelty of the proposed system.

IV. THE PROPOSED FRAMEWORK: BOOSTED SLICE CLASSIFIERS (BSC)

A. 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, each of length $N_{\rm X}$. Let $\mathbf{X} = [X(1), \cdots, X(N_{\rm X})]^T$ be an instance of such a spectral vector. In particular, let $\mathbf{X}_j$ denote the j-th vector in the sequence. The slice feature L_i is calculated from $\mathbf{X}$ as follows:

$$L_i \;\equiv\; L_i(\mathbf{X}) \;= {\rm X}(k_{i,1}) - {\rm X}(k_{i,2}). \qquad (1)$$

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 $N_{\rm X}$ but cannot be equal, restricting the total number of slice features as defined above to $N_L = N_{\rm X}(N_{\rm X} - 1)$. Let $L_i(\mathbf{X})$ be denoted by L_i for short.

B. Feature modelling: Slice Classifiers

Each slice L_i has a slice classifier f_i associated with it. 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 client ('1') or impostor ('0'). The output of f_i is calculated as,

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

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 the total number of slices, N_L.

C. 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 done by the Discrete Adaboost algorithm [14] widely used for such classifier selection tasks [11] [12]. In this algorithm, the positive training samples are extracted from the speech data of the specific client, while the negative samples come from a general pool of speakers, distinct from the client. The same pool of 'impostor' speakers termed the "world" set are used for *all* clients, as in the Universal Background Model (UBM) framework for standard speaker verification systems (ref. Sec. II).

This algorithm works in a loop. In each iteration of the loop, it selects one slice classifier out of the set based on how well it performs on a subset of training samples. Each iteration has three parts: 1) Selecting a fixed number N^*_{tr} of training samples based on their weights. Samples with higher weights are more likely to be selected.[5] Usually N^*_{tr} is a small fraction of the total number of training samples, N_{tr}. 2) Selecting the best performing slice classifier trained on this subset of training samples. 3) Classifying *all* the training samples using this selected best slice classifier and re-weighting all of the samples, so that misclassified samples' weights are proportionately increased, and correctly classified ones' weights are decreased. In addition to selecting the slice classifier, each iteration also assigns a weight to the selected classifier, based on its efficiency.

We note that classifier selection and feature modelling (classifier training) are linked and happen alternately. The re-weighting of training samples based on prior classification performance serves as the feedback link between classifier selection and feature modelling or classifier training: subsequent classifiers are selected based on their ability to classify samples which were poorly classified by previously selected classifiers. This is a novel concept not found in standard SV systems. Also, due to the re-weighting procedure, misclassified samples get more weight in successive iterations. This implies that, in effect, the more confusable speakers in the impostor set are expected to get more importance, analogous to the idea of *cohorts* in the standard approach [23]. However, in this case, the distinction between what is more easily- and less easily-classifiable is at the frame level, not at the speaker level.

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 ($N^*_{tr} \ll N_{tr}$) [6].
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 probability distribution given by weights $\{w_{n,j}\}$
   • From this subset, extract all the slice features $\{L_i\}_{i=1}^{N_L}$ (ref. Eq.1). For each slice feature $L_i$, train a threshold classifier $f_i$. Let misclassification error of $f_i$ be $\epsilon_i$.
   • Select the next best slice classifier, $f^*_n = f_{i^*}$ and its associated slice $L^*_n = L_{i^*}$ where $i^* = \arg\min_i \epsilon_i$, ie. the classifier with the lowest error. Let $\epsilon_{i^*}$ be the misclassification error of the selected slice classifier.
   • Set $\beta_n \leftarrow \frac{\epsilon_{i^*}}{1-\epsilon_{i^*}}$.
   • 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 $\{\alpha_n\}_{n=1}^{N^*_L}$.
```

D. Slice classifier combination

For each client, the selected slice classifiers are combined via a linear weighted sum F which is called a strong classifier [14]. Let $\mathbf{X}$ be a test spectral vector extracted from an utterance, U. Then the strong classifier score is calculated as,

$$F = \sum_{n=1}^{N^*_L} \alpha_n f^*_n(L^*_n(\mathbf{X})). \tag{3}$$

Scores from each frame in the utterance are added and normalized by number of frames, to obtain the final score for the utterance. This is compared with a preset threshold to decide if the utterance was made by a client or an impostor. This preset threshold Θ is set to correspond to the Equal Error Rate [4].

V. Experimental Validation - Brief Overview

To validate the effectiveness of the proposed BSC framework in view of the two objectives mentioned in Section I, ie. robustness and computational efficiency, two groups (A and B) of speaker verification experiments were carried out, with different levels of difficulty:

Group A: Experiments were carried out on easy to moderately challenging databases. The proposed framework was compared with baseline MFCC/GMM-UBM reference systems (ref. Sec.II). The experiments were carried out for each of the following conditions:

1) Experiments on clean speech (Sec.VI-A). The database used was TIMIT [16].
2) Experiments on noisy speech. Two different noise classes were considered:
    a) Additive noise (Sec.VI-B). Database used was TIMIT. Three types of noise (white, pink and babble) at SNRs ranging from 5dB to 20dB were added *only* to the test segments.

[5] Initially, the sample weights are all uniform.

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

b) Convolutive noise (Sec.VI-C). Database used was HTIMIT [17]. Eight different microphone types were considered while testing.

These experiments involved a mismatch between training (using only clean speech) and test (using noisy speech). However, this mismatch was *artificially* induced in the data.[7]

Group B: Experiments were carried out on a much more challenging recent database (MOBIO [2] [24]) (Sec.VII). The proposed framework was compared with multiple state-of-the-art reference systems (Sec.II) unlike only baseline systems as in Group A. These experiments involved speech data captured on mobile phones and there was mismatch at multiple levels in the data. This mismatch was naturally created as a direct consequence of the recording scenario, in contrast to Group A.

Each of these experiments are detailed further in the following sections.

VI. Group A Experiments

This group of experiments involved easy to moderately difficult conditions.

A. Experiments on clean speech: matched condition

The main aim for this first set of experiments was to provide an initial proof-of-concept and demonstrate the feasibility of the proposed system. These experiments evaluated how well our system can perform *text-independent* speaker verification with large populations under near-ideal conditions, compared to a baseline GMM-UBM reference system.[8]

1) Database description: The TIMIT database was chosen for this part of the work [16]. It is a standard database with no intersession variability, acoustic noise or microphone variability [23]. Each utterance is a read sentence of approximately 3 seconds duration. The training and test sentences have different lexical content, hence this is an example of text-independent speaker verification. The sampling frequency is 16kHz.

2) Systems evaluated, protocol and experimental details: To compare the proposed BSC system, the standard MFCC-GMM system detailed in [23] was chosen as reference. The speaker verification protocol as used by Reynolds et al. in [23] was followed. The 168 speakers (112 males, 56 females) from the "test" portion of the TIMIT database were used as clients. For each speaker, the 2 *sa* sentences, 3 *si* sentences and first 3 *sx* sentences were used for training and the remaining 2 *sx* sentences for testing.

For all systems, speech was segmented into frames by a 20 ms window progressing at a 10 ms frame rate [23]. For the BSC system, each frame was processed by a 256-point FFT. One half of the symmetric magnitude spectra was retained to form the spectral vectors $\mathbf{X}$ of length $N_{\mathrm{X}} = 128$.

For training a client classifier in the BSC system (ref. Secs.IV-B, IV-C), the positive (client, '1') training samples were extracted from the client training data, while the negative (impostor, '0') samples were extracted from a set of 250 utterances randomly selected from the "train" portion of the TIMIT database. These utterances were made by speakers *all distinct* from the "test" portion, and the *same* negative samples were used for *all* the clients. We term this as the "world" set (ref. Sec.IV-C).

As mentioned earlier, the BSC system was compared with a standard MFCC-GMM system [23]. For clarity, we describe here only the chief aspects of this system. Firstly, 12$^{\text{th}}$ order Mel-frequency cepstral coefficients (MFCC) were extracted from the speech frames [25] [26]. These were then modelled by 32-mixture GMM [23]. To model impostors, each client had its own specific "world" or "background" set[9] of speakers, selected from the set of clients itself [23]. Depending on the selection criterion of the background speakers, two reference system configurations were considered, namely

1) Reference system TI: 10 "maximally spread close" (msc) background speakers were selected.
2) Reference system TII: 5 msc + 5 "maximally spread far" (msf) background speakers were selected.

During testing, the mean log-likelihood of the 10 background set models is subtracted from the log-likelihood of the claimed client model [23] to estimate the log-likelihood ratio score of a test utterance.

For evaluating the BSC system, experiments were performed using each of the 168 speakers acting as the claimant, with each of the remaining 167 speakers acting as impostors, and rotating through all speakers. Since the negative samples in training came from a distinct "world" set, all the remaining 167 speakers were treated as impostors. For testing the reference systems TI and TII, the same experiments were performed as for the BSC system, excluding the 10 background speakers for each client from the impostors because these systems did not use a single distinct "world" set as the BSC system.

Experiments were conducted separately for three conditions [23]:

1) Mixed sex (F+M), using all 168 speakers.
2) Male only (M) (112 speakers).
3) Female only (F) (56 speakers).

The performance of each system was measured in terms of the global Equal Error Rate (EER) computed using a client-independent threshold [23] on test data. For this, the threshold Θ (ref. Sec.IV-D) at which the false-acceptance (FA) rate equals the false-rejection (FR) rate is calculated, considering *all* client and impostor test scores together, and the FA using this threshold is reported as the EER.

Thus, the global EER measures the overall (client-independent) performance of the system and is likely to be much more statistically significant than results based on client-dependent thresholds [23].[10] We did not re-implement the

[7] This was done by either adding the noise signal to the clean speech, or by playing the original speech and recording it by different microphone types.

[8] This is an extension of previous studies on the XM2VTS database [15] to the text-independent case. These previous studies had shown that the proposed system performed well but were limited by the fact that the lexical content in training and test were the same, ie. it was not known how the system could perform in the case of text-independent SV.

[9] This is alternatively termed the "cohort" set [4].

[10] Henceforth, we shall use EER to mean *global* EER.

reference systems. We report here directly the results of the systems from [23].

3) Results: In all the three experimental conditions, the proposed BSC system has performed equally well as or very close to the reference systems. The EER of the systems have been shown in Figure 2 (a)-(c). For the BSC system, the EER has been plotted against the number of slice classifiers N_L^* selected by Adaboost and used to form the final strong classifier F (ref. Sec.IV-D).

The EER of the BSC system drops quickly from above 5% below 10 slice classifiers to below 1% after about 250 slice classifiers have been selected. For all 3 conditions, the EER consistently shows a downward trend with increasing N_L^*, interspersed with small oscillations, finally reaching a saturation level.[11]

This saturation level is close to the EER achieved by reference system TII for all 3 cases. For the F+M case, it is slightly lower than the TI EER while for the M only and F only cases, it is slightly higher than the TI EER. This saturation level is reached after about 400 to 450 slices have been selected. At this value of N_L^*, the computational complexity of the BSC system is significantly lower than the reference systems.

B. Experiments on speech corrupted by additive noise: mismatched condition

The aim here was to examine the effect of mismatched additive noise on the performance of our proposed system, compared to a standard reference system.

1) Database description: The TIMIT database [16] was used in this part of the work also. The training and test sentences had different lexical content, hence this is also an example of text-independent speaker verification. However, the original clean TIMIT data was used only for training. For testing, TIMIT data corrupted by additive noise was used. For this, three types of noise from the Noisex-92 database [27], namely, white, pink and babble, were added to each test utterance at 4 SNR levels (20dB, 15dB, 10dB and 5dB).[12]

2) Systems evaluated, protocol and experimental details: Apart from the proposed system, a standard MFCC-GMM system [4] was used as reference.[13] The BSC system configuration was precisely the same as in Sec.VI-A. For the reference MFCC-GMM system, we experimented using different number of cepstral coefficients (12 and 16) [4] [26] for the features and different number of Gaussians (from 32 to 1024) for the GMM.

Instead of the client-dependent background set described in Sec.VI-A and [23], we used a common impostor set to create a single GMM model called the Universal Background Model (UBM) to model impostors (Section II). The advantage for large speaker databases is that individual background sets need not be selected for each client.[14]

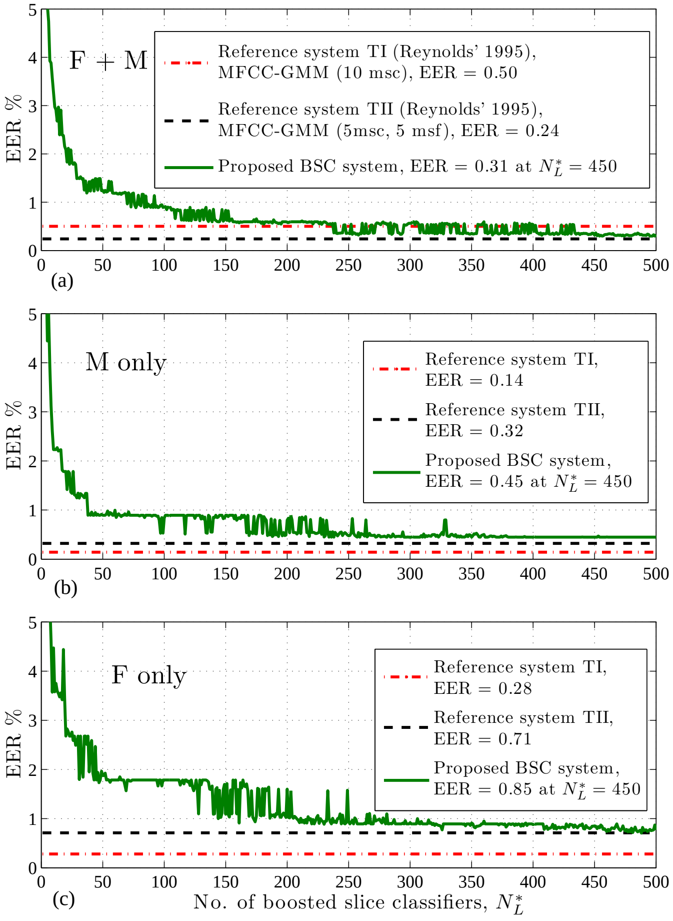


Fig. 2. Equal Error Rates (EER %) for same-sex (M only, F only) and mixed-sex (F+M) experiments on the TIMIT database, for the proposed BSC system and two MFCC-GMM based reference systems TI and TII. For the BSC system, EER has been plotted *vs* N_L^*, the number of boosted slice classifiers used to form the strong classifier F. The numerical values of the EERs are shown in the legend boxes. For the BSC system, the EER at a particular point $N_L^* = 450$ is shown. The reference systems are from Reynolds [23]. Please consult the text (Sec.VI-A) for more details.

For fair comparison, this common impostor set is the same as the "world" set which provided the negative samples for the BSC system (ref. Sec.VI-A2) extracted from the "train" part of TIMIT. For each client, a model is created by adapting the means in the UBM using the client training data [4].

Among the different configurations of reference systems tried, we report here the two overall best performing ones:

1) Reference system NTI: 16 MFCC + 16ΔMFCC + ΔEnergy, Cepstral Mean Subtraction (CMS) [4] and 1024-mixture GMM.
2) Reference system NTII: 12 MFCC [25] [26], no delta, no CMS, and 1024-mixture GMM.

The features used by the system NTII are the same as the reference systems in Sec.VI-A [26] while the features used by system NTI involve slightly more calculations [4].

The same speaker verification protocol as used in Sec.VI-A was followed [23]. The experimental details were exactly the same as in Sec.VI-A except for one difference: all the data

[11] Although only the first 500 slices are shown in Figure 2, we conducted experiments using upto 1000 slices and the test EER still remained stable.

[12] The noise segments were randomly chosen and were equal in length to the test segments.

[13] In this case, we implemented the reference system ourselves.

[14] The single background model has become the predominant approach used in speaker verification systems [4].

for training came from original TIMIT while for test, different types of noise from the Noisex-92 database was added to it. Apart from this, in this experiment, *all* the remaining 167 speakers were used as impostors even for the reference systems since they used the distinct “world” impostor set for training, like the BSC system.

Separate experiments for the 3 different noise types at 4 different SNR levels were conducted, leading to 12 different conditions. In the face of this, experiments were conducted for mixed sex (F+M) condition only, using all 168 speakers [23]. The performance of the systems was calculated in terms of the global equal-error rate (EER) as before.

3) Results: For all the 3 noise types and 4 SNR levels, the proposed BSC system has performed equally well as or better than the reference systems. The EER of the systems have been shown in Figure 3 (a)-(l). For the BSC system, the EER has been plotted *vs.* the number of slice classifiers N_L^*. As in Sec.VI-A, the EER of the BSC system has shown a general downward trend with increasing N_L^* although the errors are much higher here due to the more difficult testing scenario.

For pink and babble noise, the EER has either continued dropping or saturated at a certain level, without any subsequent increase (Figures 3 (e)-(l)). For white noise (Figures 3 (a)-(c)), the BSC system EER has increased slightly at some points. In spite of this, for both white and babble noise, the BSC system has outperformed the reference systems much before $N_L^* = 100$. For pink noise, the BSC system has consistently outperformed system NTII while it finally catches up with system NTI in all cases.

We note that these results support the evidence of previous studies by the authors [15] where a similar framework involving boosted binary features performed better than the standard MFCC-GMM system on speech corrupted by different types of additive noise.

C. Experiments on speech corrupted by channel noise: mismatched condition

The aim here was to examine channel effects, more precisely handset transducer effects, on the performance of our system, compared to a standard MFCC-GMM system.

1) Database description: The handset TIMIT (HTIMIT) database was chosen for this work [17]. The database was constructed by playing a gender-balanced subset of the TIMIT database through a Sennheizer head-mounted microphone (‘senh’) and 8 telephone headsets: 4 carbon button microphones (‘cb1’-‘cb4’), 4 electret microphones (‘el1’-‘el4’) and one Sony portable microphone (‘pt1’). In this way, headset transducer degradations were imposed in a systematic way, keeping the speaker and linguistic richness of the original TIMIT database. The training and test sentences had different lexical content, hence this is also an example of text-independent speaker verification.

2) Systems evaluated, protocol and experimental details: Apart from the proposed system, the MFCC-GMM system described in Sec.VI-B2 [4] [28] was used as reference.[15] To reduce linear filter effects due to the headset transducers, cepstral mean subtraction (CMS) was performed on the MFCC for the reference system. Similarly, for the BSC system, the spectral magnitude vector $\mathbf{X}$ (ref. Sec.IV-A) was replaced by its $\log$ followed by mean normalization.

As in Sec.VI-B2, different values of the metaparameters (ie. number of cepstral features, number of Gaussian mixtures) were tried for the reference system. Among the different configurations tried, we report here two of the best performing ones:

1) Reference system HTI: 16 MFCC, CMS and 32-mixture GMM.
2) Reference system HTII: 16 MFCC [25] [26], CMS, and 1024-mixture GMM.

The speaker verification protocol for HTIMIT described in [28] [29] was taken as a guideline. More precisely, 100 speakers were randomly chosen out of the total 384 to form the client set. A different subset of 50 speakers were randomly chosen as the test impostor set. In addition, 250 randomly chosen utterances from the remaining speakers were used as the “world” set during training (ref. Secs.VI-A2 and VI-B2). All sets were gender balanced.

For each client, the 2 *sa* and 5 *sx* sentences recorded using the ‘senh’ microphone only were used for training. For testing, separate experiments were performed using the 3 *si* sentences recorded using the ‘senh’ microphone and *all* the 8 headset types. We note that this consists of one matched condition (‘senh’-‘senh’) and 8 mismatched conditions (‘senh’-‘cb1’,‘senh’-‘cb2’,$\cdots$, ‘senh’-‘pt1’).

Each client model was tested against its own 3 *si* sentences (3 true accesses) and the 3 *si* sentences of all 50 speakers in the test impostor set (150 impostor accesses). This was repeated for all 100 clients. The performance of the systems was calculated in terms of the global equal-error rate (EER) as before, for each microphone type separately.

3) Results: For all the 9 conditions tested, the proposed BSC system has performed nearly as well as the reference systems. The EER of the systems have been shown in Figure 4 (a)-(j). For the BSC system, the EER has been plotted *vs.* the number of slice classifiers N_L^*.

As before, the EER of the BSC system has shown a general downward trend with increasing N_L^* and saturates to values around 10% for *all* the 9 conditions. It is noteworthy that the performance of the proposed system is fairly independent of the microphone type.

On the contrary, the reference systems have shown a wider variation in EER, particularly if we observe their performance for ‘senh’ and ‘cb3’.[16] This is an important contrast between the proposed and reference systems. Also, there is no single best reference system: for some microphones HTI is better than HTII while for others, it is the reverse.

VII. Group B Experiments

This group of experiments involved more difficult conditions and were performed on the MOBIO database.

[15] In this case also, we implemented the reference system ourselves.

[16] These two microphones have the best and worst sound characteristics respectively [17].

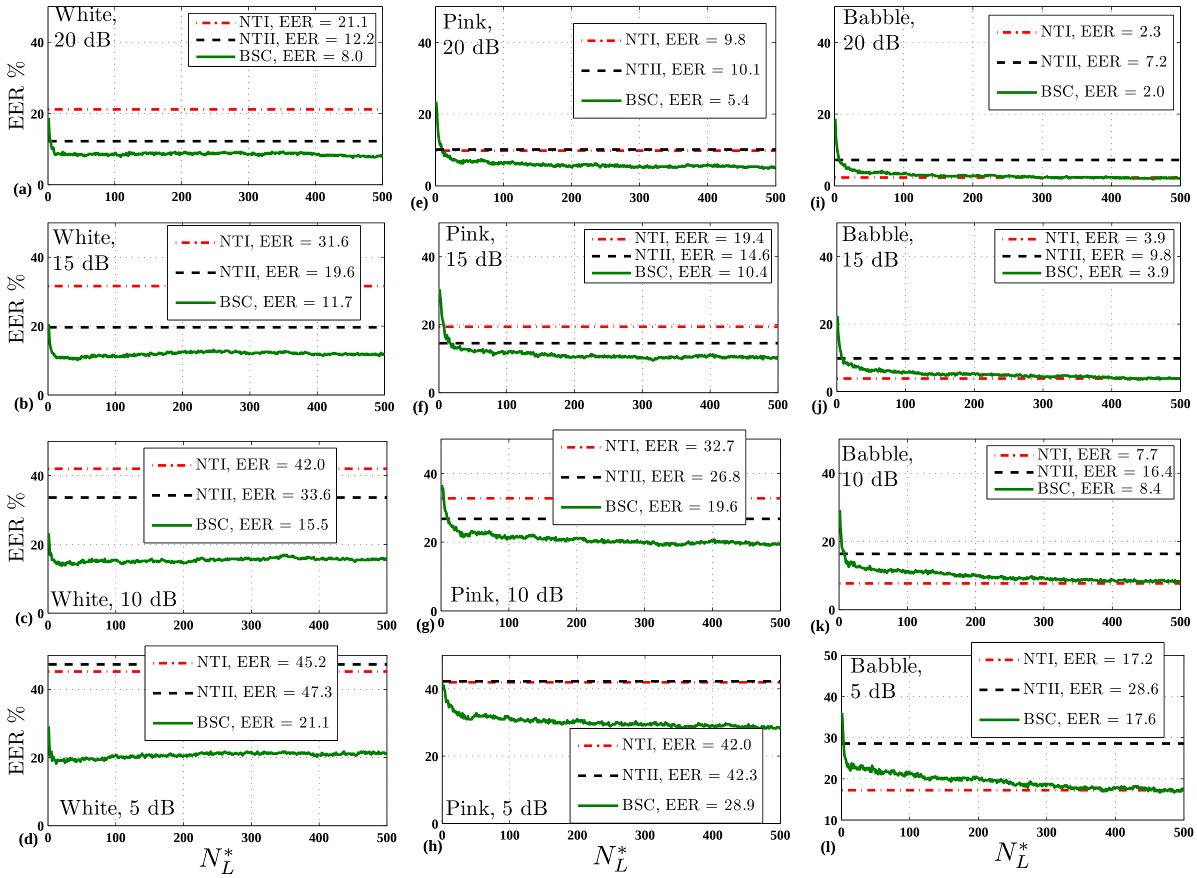


Fig. 3. Equal Error Rates (EER %) for mixed-sex (F+M) experiments on the noisy TIMIT database (TIMIT + Noisex), for the proposed BSC system and two MFCC-GMM based reference systems NTI and NTII. For the BSC system, EER has been plotted *vs* N_L^*, the number of boosted slice classifiers used to form the strong classifier F. Three different noise types at four different SNR levels have been considered. The noise type and level are shown in each subfigure. The numerical values of the EERs are shown in the legend box. For the BSC system, the EER at $N_L^* = 450$ is shown. Please consult the text (Sec.VI-B) for more details.

A. Database description

Experiments were performed on the MOBIO Phase I database [1] [2] 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. 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 a mobile display), b) one question requiring one 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 was chosen because it has some challenges. Firstly, all speech data was collected on mobile phones and had significant amount of noise [1]. About 10 % of the utterances had SNRs less than 5 dB, while 60 % had SNRs between 5 to 10 dB. Secondly, 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. Thirdly, the data presented possibilities for testing different levels of mismatch using a challenging protocol (Section VII-C). Also, it was used for the recent MOBIO Face and Speaker Verification Evaluation contest at ICPR 2010.[17] Hence, there already exists a large number of reference results from various sites involving state-of-the-art SV systems. This is useful for comparison.

B. Systems evaluated

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).[18] All of these participated in the MOBIO evaluation at ICPR 2010.

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

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

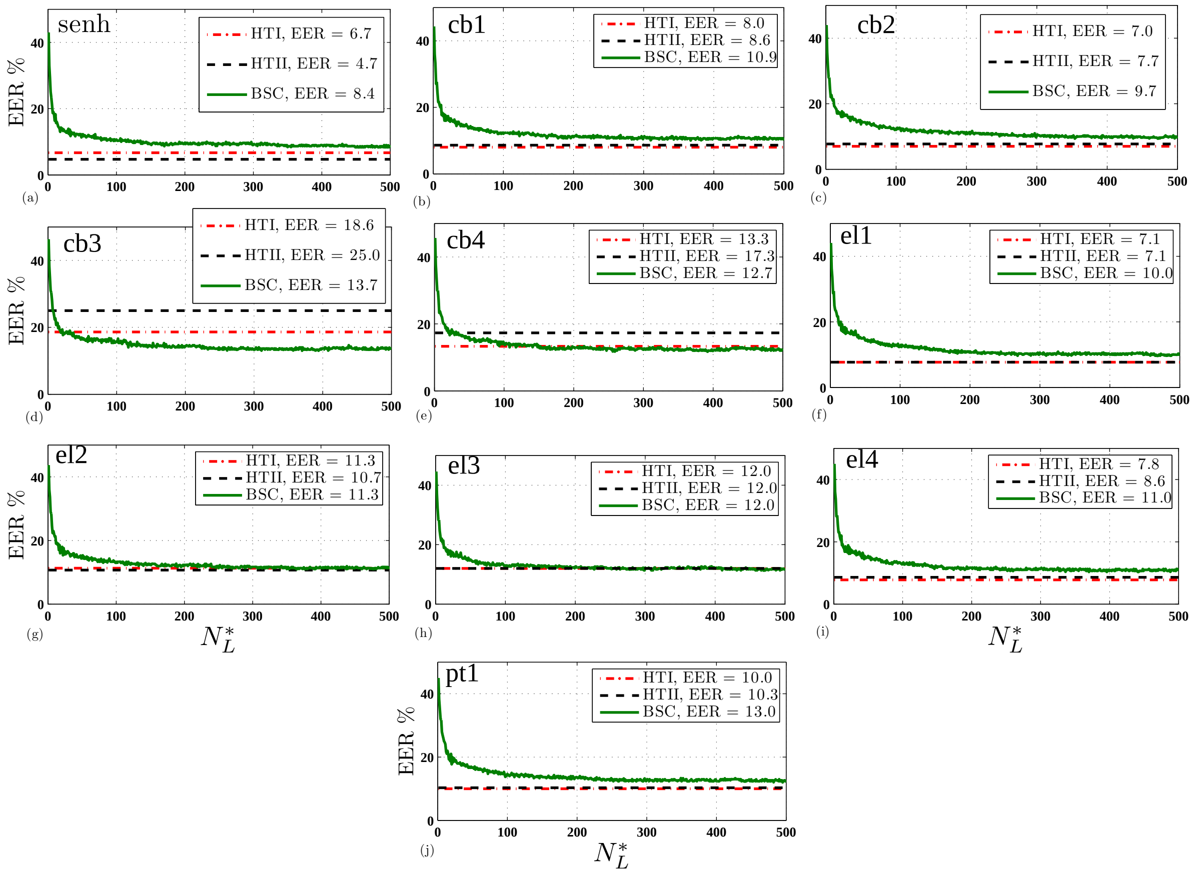


Fig. 4. Equal Error Rates (EER %) for mixed-sex (F+M) experiments on the HTIMIT database, for the proposed BSC system and two MFCC-GMM based reference systems HTI and HTII. For the BSC system, EER has been plotted *vs* N_L^*, the number of boosted slice classifiers used to form the strong classifier F. Ten different microphone types have been considered. Training for all systems was done using only data collected by the Sennheizer (senh) microphone. The numerical values of the EERs are also shown in the legend box. For the BSC system, the EER at $N_L^* = 450$ is shown. Please consult the text (Sec.VI-C) for more details.

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 I
BASIC PARAMETERS OF THE REFERENCE SYSTEMS, GROUPED ACCORDING TO SUBMITTING INSTITUTION. PLEASE SEE SEC.VII-B FOR DETAILS.

The details of these systems are provided in [1] [2]. For convenience, we highlight here the chief aspects of these reference systems. All reference systems used cepstral features [4]. 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 type 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 VII-B). 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 I-vector 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, precisely the same setup as in previous experiments was used (Sec.VI-A2). No processing step was changed or added nor were any system parameters tuned.

C. Protocol and experimental details

The SV protocol used was the same as in the MOBIO Evaluation, details of which are given in [1] [2]. 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 distinct in terms of speakers and data collection sites. 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.

The protocol for MOBIO presents some special challenges in addition to the noisy data itself. They are: 1) *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. 2) *Lexical mismatch* Speech used in enrolment and testing had different lexical content (*text-independent* SV task). 3) *Speech-type mismatch.* The training (enrolment) was done on read speech while testing was on free speech.4) *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.

D. Results

The Half Total Error Rate (HTER %) on the test set of the MOBIO database for all the 18 systems have been shown in Figure 5. In all cases, the performance of the proposed BSC system is reasonably good, lying near the mean of the reference systems’ performance. It is noteworthy that the proposed system achieved reasonable performance using only a simple framework involving a weighted sum of threshold-based classifiers. For both genders, the BSC system performance saturated around $N_L^* = 100$. In contrast, most of the reference systems used more complex techniques such as SVM supervectors and factor analysis in addition to standard MFCC-GMM setup.

While such enhancements enabled the best reference systems to perform better than the proposed BSC system, several of the reference systems also performed worse than BSC in spite of their complexity. This indicates that the BSC system achieves a good trade-off between system performance and computational complexity, consistent with results from experiments in previous sections.

VIII. Analysis of Proposed System

From Secs.VI-A3, VI-B3, VI-C3 and VII-D, we observe that the proposed BSC system shows comparable text-independent speaker verification performance vis-à-vis the standard SV systems (both baseline and state-of-the-art) across a wide spectrum of conditions, both clean and noisy, matched and mismatched, using speech either collected using a standard microphone setup or a mobile phone.

Hence, it seems to fulfill the first objective outlined in Section I, ie. robustness. At the same time, the proposed system fulfills the second objective, ie. computational efficiency. In the next two sections, we analyze these two important aspects

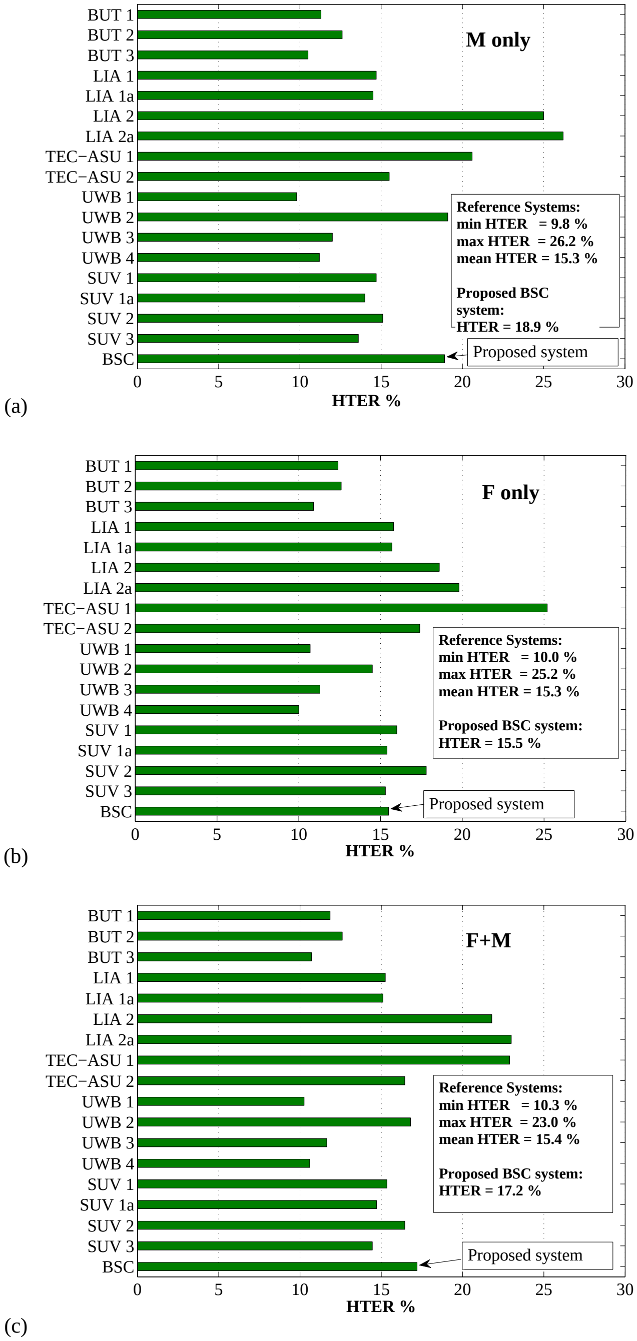


Fig. 5. Half Total Error Rates (HTER %) for SV experiments on the Test set of the MOBIO Phase I database using (a) only male speakers, (b) only female speakers and (c) and average of the two. HTERs are shown for the proposed BSC system and 17 reference systems. Please consult the text (Sec.VII) for more details.

of the proposed system: a) robustness in the presence of noise (Section VIII-A), and b) computational complexity (Section VIII-B). In the final section, we analyse the distribution of the slice classifiers selected by the system as being the most discriminative.

A. Robustness to additive noise

In Sec.VI-B3, it was shown that the BSC system was significantly more robust to different types of additive noise in a mismatched scenario, than the MFCC-GMM system. This is an important property of the BSC system. Here, we present an analysis of this property at the frame level.[19]

For our analysis, we picked out two speakers from the TIMIT database at random. The first speaker served as the true client, while the second served as an impostor.[20] We had already created the models for the client (the strong classifier F for the BSC system and the UBM-GMM for MFCC-GMM system) using clean training data. Next, one speech frame from the test data of both speakers in the TIMIT database was extracted. Three types of noise (white, pink and babble) at 4 different SNRs were subsequently added to these clean speech frames to create noisy speech frames (ref. Sec.VI-B). These frames were then passed to the client models and finally the frame scores were generated.

The process of score generation is depicted in Figure 6. In this figure, the left half illustrates true client accesses, ie. the client speech frame was matched with the client model, while the right half illustrates impostor accesses, ie. the impostor speech frame was matched with the client model. The first three rows from the top depict the BSC system while the last two depict the MFCC-GMM system.

1) Frame-level behaviour under clean condition: In the first row, subfigures (a) and (c) show the outputs of the first 40 boosted slice classifiers $\{f_n^*\}_{n=1}^{40}$ of the BSC system for the clean speech frames. We note that the classifiers are predicting mostly 'client' ('1': light yellow bands) for the client frame and mostly 'impostor' ('0': dark green bands) for the impostor frame. The precise number of classifiers predicting '1' is shown in subfigures (e) and (g) in the second row: a high number for the client and low for the impostor. The final scores $F(\mathbf{X})$ considering only these first 40 classifiers (ref. Eq.3) is shown by the green broken line in subfigures (i1-3), (j1-3).

In the fourth row, subfigures (k) and (m) show the cepstral vectors X_M extracted from the clean speech frames for the MFCC-GMM system. The loglikelihood ratio scores $S(X_M)$ obtained by passing these vectors through the UBM-GMM of the client is shown by green broken line in subfigures (o1-3), (p1-3).

For both the BSC and MFCC-GMM systems, we see that the client and impostor scores are well separated in the clean condition.

2) Frame-level behaviour under noisy condition: In the case of the BSC system, as different types of noise are added at different SNRs to the clean test frame, the slice classifier outputs vary due to the change in the shape of the spectrum. These variations $\{\Delta(f_n^*)\}_{n=1}^{40}$ are shown in subfigures (b1-3), (d1-3). The main point to note is that a significant number of slice classifier outputs remain *unchanged* after noise addition, ie. $\Delta(f_n^*) = 0$. These are marked by light green bands. Several classifier outputs change from '1' (correct) to '0' (error) for the client frames ($\Delta(f_n^*) = -1$, dark green band), and '0' (correct) to '1' (error) for the impostor frame ($\Delta(f_n^*) = 1$, yellow band). However, the error is limited exclusively to these outputs. Interestingly, some erroneous outputs become correct too ($\Delta(f_n^*) = 1$ for the client and $\Delta(f_n^*) = -1$ for the impostor).

The number of classifiers predicting '1' is again shown in subfigures (f1-3), (h1-3) and the final scores $F(\mathbf{X})$ are shown by the red lines in subfigures (i1-3), (j1-3). We note that the client and impostor scores have approached each other gradually, as the SNR has reduced. This is expected.

Similarly, in the case of the standard MFCC-GMM SV system, as different types of noise are added to the clean test frame, the cepstral vectors X_M change values. These changes $\Delta(X_M)$ are shown in subfigures (l1-3), (n1-3).[21] Contrary to the BSC system where the error is limited to certain slices, we note that the entire cepstrum has been distorted by noise, even when the SNR is high. Some cepstral coefficients will be affected more and some affected less. The loglikelihood ratio scores obtained by passing these noisy vectors through the UBM-GMM of the client is shown by the red lines in subfigures (o1-3), (p1-3).

We observe that for each noise type and SNR level, the client and impostor scores have approached each other *less* in the BSC system than in the MFCC-GMM system, which would lead to better separability and lower verification errors for the BSC system. This is mainly due to the fact that although the noise did affect some of the slice classifier outputs, it could not affect a large fraction of the outputs. These correct outputs could combine together and offset the effect of the incorrect ones. In MFCC-GMM system, the entire cepstrum is affected and we cannot avail of this unique advantage, which is a characteristic of parts-based systems.

B. Complexity of the system

In this section, we compare the computational complexity of the proposed BSC system with that of the reference systems [30]. For simplicity, we consider only the reference systems used in Group B experiments (ref. Sec.VII). We consider 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 until the calculation of the final score at a frame level. In fact, the BSC system has a simpler feature extraction stage, with no filterbanks nor feature warping. For the sake of simplicity, we ignore this.

[19] For simplicity, we restrict ourselves to analysis of system behaviour under additive noise in this work. Similar analyses could be carried out for the case of convolutive noise also.

[20] We shall henceforth denote them as 'client' and 'impostor' respectively.

[21] The same noisy frame was used for both the BSC and MFCC-GMM systems, for all cases.

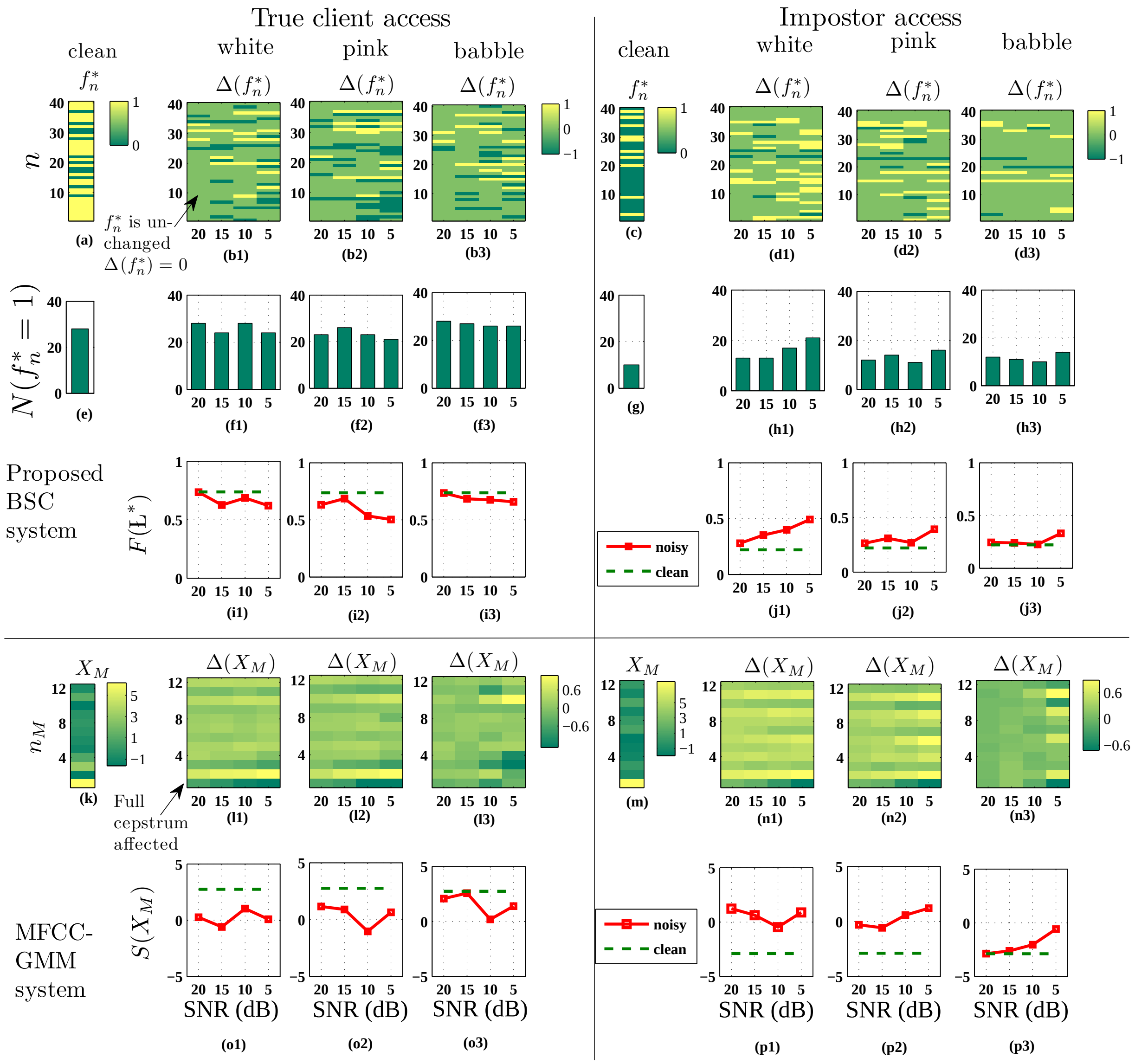


Fig. 6. Effect of additive noise: (a, c) Outputs of the first 40 boosted slice classifiers $\{f_n^*\}_{n=1}^{40}$ using clean speech frames. (b1-3, d1-3) Changes $\{\Delta(f_n^*)\}_{n=1}^{40}$ in the classifier outputs as 3 different noise types are added to the speech frames at 4 SNRs. (e, f1-3, g, h1-3) Number of classifiers with output $f_n^* = 1$ for each of the above cases. (i1-3, j1-3) Final strong classifier output F for the above cases. (k, m) MFCC vectors X_M extracted from the same clean speech frames as in (a, c). (l1-3, n1-3) The changes $\Delta(X_M)$ in the MFCC vectors due to additive noise. (o1-3, p1-3) Loglikelihood ratio scores using these MFCC vectors for the standard MFCC-GMM system. Please see the text (Sec.VIII-A) for more details.

1) Reference MFCC-GMM system: For reference systems, we consider only the essential modelling block while computing the number of FLOPs, ie. only the computation of the Gaussian components for GMM-based systems. We ignore all other blocks, such as those related to factor analysis, I-vector, 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.

Let N_D be the feature dimension of the MFCC feature vector $\mathbf{X}_M$ extracted from one frame of speech. To evaluate a single Gaussian, $G(\mathbf{X}_M; \mu, \mathbf{\Sigma}, p)$ with mean vector μ, diagonal covariance matrix $\mathbf{\Sigma}$ and mixture weight p using,

$$\begin{aligned} G(\mathbf{X}_M; \mu, \mathbf{\Sigma}, p) &= \frac{p}{(2\pi)^{\frac{N_D}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(\mathbf{X}_M - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{X}_M - \mu)} \\ &= \hat{p} \cdot e^{\sum_{i=1}^{N_D} (X_M(i) - \mu(i))^2 \cdot \hat{s}_i} \qquad (4) \end{aligned}$$

where $\hat{p} = \frac{p}{(2\pi)^{\frac{N_D}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}}$, $\hat{s} = -\frac{1}{2\sigma(i)^2}$ and $\{\sigma(i)\}_{i=1}^{N_D}$ are the diagonal elements of $\mathbf{\Sigma}$ (which can all be precomputed), the number of floating point additions, multiplications and exponentiations involved are $2N_D - 1$, $2N_D + 1$ and 1 respectively.[22] However, most practical GMM implementations involve code optimization, which reduces the number of

[22] We note the replacement of division by multiplication (with $\hat{s}$) in Eqn. 4 because multiplication is usually faster than division [31].

FLOPs. In particular, the exponentiation can often be avoided by the `log-add` operation.

Hence, in order to keep the current analysis simple, again at the cost of a pessimistic bias against our system, we only consider the computation of the quadratic term $(\mathbf{X}_M - \mu)^T \mathbf{\Sigma}^{-1}(\mathbf{X}_M - \mu) \equiv \sum_{i=1}^{N_D}(X_M(i) - \mu(i))^2 \cdot \hat{s}_i$ in Eq. 4. This term *must* be computed once per Gaussian, independent of the level of optimization achieved. To compute it, $2N_D$ floating point multiplications and $2N_D - 1$ floating point additions are required. Hence, to process one frame of speech, we multiply these quantities by N_G, the number of Gaussians. Thus, the total number of multiplications and additions per frame is, $n^{\times} = 2N_G N_D$, $n^{+} = N_G(2N_D - 1)$ respectively. Hence, the total number of FLOPs per frame is:

$$N_{\text{FLOP}} = n^{\times} + n^{+} = N_G(4N_D - 1). \tag{5}$$

2) Proposed BSC system: Let $\mathbf{X}$ be a spectral vector extracted from a speech frame (ref. Sec.IV-A). Let N_L^* be the number of slice classifiers used to form the strong classifier F (ref. Sec.IV-C). To obtain the final frame-level score $F(\mathbf{X})$, we must use Eqns. 1, 2 and 3 which we combine and implement as follows:

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

Here, $a[b]$ denotes the b-th element of array a. Since they usually take almost the same time [31], we group the number of comparisons, additions and subtractions as n^{+}. From the above implementation, we find that for the BSC system, no multiplication is required and,

$$N_{\text{FLOP}} = n^{+} = 3N_L^* \tag{6}$$

The total number of FLOPs for BSC and reference systems calculated using Eqns. 5 & 6 are shown in Fig. 7. Parameter values for N_D, N_G in Eqn. 5 are enlisted in Table I. In Eqn. 6, parameter $N_L^* = 100$.[23]

It is observed from Fig. 7 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 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 (ref. Sec. I).

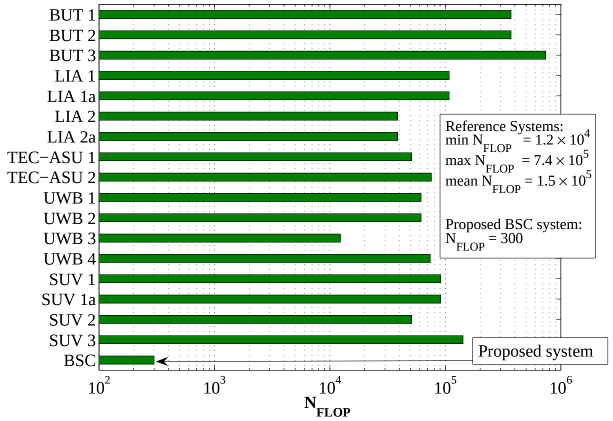


Fig. 7. No. of floating-point operations, N_{FLOP} plotted in log-scale, for the 17 reference systems used in the Group B experiments and the proposed BSC system. Please see text (Sec. VIII-B) for more details.

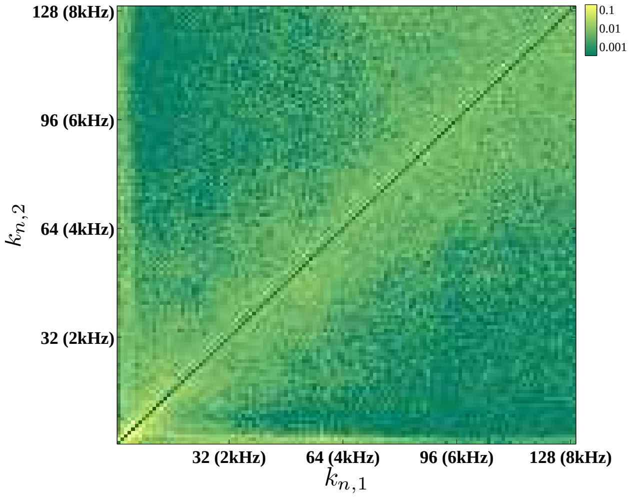


Fig. 8. Distribution of slices selected by Adaboost: The image intensity at a point $\{k_{n,1}, k_{n,2}\}$ in the image indicates the expected weight of the slice classifier corresponding to the slice defined by the parameters $\{k_{n,1}, k_{n,2}\}$, as set by the Adaboost algorithm (ref. Sec. VIII-C).

C. Distribution of selected slice classifiers

We analyse the distribution of slice classifiers which were selected by Adaboost. In Figure 8, we show the matrix of expected weights assigned to all the slice classifiers, indexed by their frequency points $\{k_{n,1}, k_{n,2}\}$, averaged over all 168 clients in the TIMIT database. A higher weight (indicated by lighter yellow colours) would mean a more discriminative or speaker-informative slice, as determined by Adaboost. From the figure, we observe that certain $\{k_{n,1}, k_{n,2}\}$ pairs have distinctly higher expected weights than others although in general, slices were selected from all regions of the spectrum (both low frequency and high frequency regions). In particular, values of $k_{n,1},\ k_{n,2} \leq 1$kHz seem to be given higher weights. Also, pairs $\{k_{n,1}, k_{n,2}\}$ with $k_{n,1}$ close to $k_{n,2}$ were given higher weights. Note that the TIMIT sampling frequency of 16 kHz was used to convert k values in the range $\{1, \cdots, 128\}$ to frequencies.

IX. Conclusions

This paper investigated a novel parts-based system for text-independent speaker verification. This system uses a novel feature called “slice feature” extracted from short-time speech

[23] This is average value at which the BSC system reached best performance (Sec.VII-D) for the Group B experiments.

spectrum and simple threshold-based “slice classifiers” selected and combined by a boosting framework. The approach was compared against standard cepstral features using both baseline and state-of-the-art speaker verification systems on TIMIT, HTIMIT and MOBIO corpus.

The proposed parts-based system showed good performance over a wide range of experimental conditions, ranging from clean speech to noisy speech collected on mobile phones. Compared to standard systems, it performed equally well in the clean condition and performed better than them in several of the noisy conditions. We note that the system configuration and parameter values of the proposed system were kept the same over *all* the conditions, unlike the standard systems.

Furthermore, the proposed approach involves lower computational complexity compared to the standard approach. 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.

Possible directions for future work include: 1) Augmentation of the feature set by joint modeling in the spectro-temporal plane (using 2-dimensional instead of 1-dimensional approach). 2) Model-level fusion between the parts-based approach and the standard cepstral feature based approach. 3) Further analysis of the precise speaker-specific acoustic information captured by the boosted slice classifiers.

Since the proposed approach involves working with specific frequency points in the spectrum, it might be directly coupled with a suitable time-frequency masking framework aimed at noise removal [32] or signal separation [33]. Finally, since the approach is data-driven, it could be applied to other related tasks like phoneme recognition. Preliminary work in this direction has shown promising results [34].

ACKNOWLEDGMENT

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] S. Marcel, C. McCool, P. Matvejka, T. Ahonen, J. Cernocký, and S. Chakraborty, “On the results of the first mobile biometry (mobio) face and speaker verification evaluation,” in *Proceedings of the 20th International conference on Recognizing patterns in signals, speech, images, and videos*, ser. ICPR’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 210–225. [Online]. Available: http://portal.acm.org/citation.cfm?id=1939170.1939200

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

[3] “MOBIO Mobile Biometry, European Funded Project (FP7-2007-ICT-1),” http://www.mobioproject.org/.

[4] F. Bimbot et al., “A Tutorial on Text-Independent Speaker Verification,” *EURASIP Journal on Applied Signal Processing*, no. 4, pp. 431–451, 2004.

[5] S.S. Chen and R. Gopinath, “Gaussianization,” in *Proc. of Neural Information Processing Systems (NIPS)*, 2000.

[6] W.M. Campbell, D.E. Sturim, and D.A. Reynolds, “Support Vector Machines using GMM Supervectors for Speaker Verification,” *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, May 2006.

[7] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, “Joint factor analysis versus eigenchannels in speaker recognition,” *IEEE Transactions on Audio Speech and Language Processing*, vol. 15, no. 4, p. 1435, 2007.

[8] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, “Score Normalization for Text-Independent Speaker Verification System,” *Digital Signal Processing*, vol. 1, no. 10, pp. 42–54, 2000.

[9] M. Turk and A. Pentland, “Face Recognition Using Eigenfaces,” in *Proc. of IEEE Intl. Conf on Computer Vision and Pattern Recognition (CVPR)*, 1991, pp. 586–591. [Online]. Available: http://www.cs.ucsb.edu/ mturk/Papers/mturk-CVPR91.pdf

[10] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[11] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 511–518.

[12] Y. Rodriguez, “Face Detection and Verification using Local Binary Patterns,” Ecole Polytechnique Federale de Lausanne, PhD Thesis 3681, 2006.

[13] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, “Fast Keypoint Recognition using Random Ferns,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 448–461, 2010.

[14] J. Friedman, T. Hastie, and R. Tibshirani, “Additive Logistic Regression: a Statistical View of Boosting,” *Annals of Statistics*, vol. 28, p. 2000, 1998.

[15] A. Roy, M. Magimai-Doss, and S. Marcel, “Boosted binary features for noise-robust speaker verification,” in *Proc. of IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 4442–4445.

[16] G. Fisher, W.M and Doddington, “The DARPA speech recognition research database: Specifications and status,” *Proc. of DARPA Workshop on Speech Recognition*, pp. 93–99, Feb. 1986.

[17] D. Reynolds, “HTIMIT and LLHDB: speech corpora for the study of handset transducer effects,” in *Proc. of IEEE Intl. Conf. on Audio, Speech and Signal Processing (ICASSP)*, vol. 2, 1997, pp. 1535–1538.

[18] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1989.

[19] J. Pelcanos and S. Sridharan, “Feature warping for robust speaker verification,” in *2001: A Speaker Odyssey Workshop*, Jun. 2001.

[20] W. Campbell, “Generalized linear discriminant sequence kernels for speaker recognition,” in *Proc. of IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2002, pp. I–161–I–164.

[21] D. Matrouf, N. Scheffer, B. Fauve, and J.-F. Bonastre, “A straightforward and efficient implementation of the factor analysis model for speaker verification,” in *Proc. of Interspeech*, 2007.

[22] N. Dehak, R. Dehak, P. Kenny, N. Brmmer, P. Ouellet, and P. Dumouchel, “Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification.” in *Proc. of International Conferences on Spoken Language Processing (ICSLP)*, Sep. 2009, pp. 1559–1562.

[23] D. A. Reynolds, “Speaker identification and verification using Gaussian mixture speaker models,” *Speech Communication*, vol. 17, pp. 91–108, 1995.

[24] “The MOBIO Database,” http://www.idiap.ch/dataset/mobio.

[25] D. Reynolds, “A Gaussian Mixture modeling approach to text-independent speaker identification,” Ph.D. dissertation, Georgia Institute of Technology, Sep. 1992.

[26] D. Reynolds and R. Rose, “Robust text-independent speaker identification using Gaussian mixture speaker models,” *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.

[27] A. Varga, H. Steeneken, M. Tomlinson, and D. Jones, “The NOISEX-92 study on the effect of additive noise on automatic speech recognition,” DRA Speech Research Unit, Technical report, 1992.

[28] K. K. Yiu, M. W. Mak, and S. W. Kung, “Environment adaptation for robust speaker verification by cascading maximum likelihood linear regression and reinforced learning,” *Computer, Speech and Language*, vol. 21, no. 2, pp. 231–246, 2007.

[29] ——, “Combining stochastic feature transformation and handset identification for telephone-based speaker verification,” in *Proc. of IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2002, pp. I–701 – I–704.

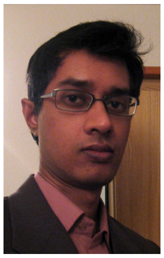
[30] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001, ch. 1: Foundations, pp. 3–122.

[31] *Intel 64 and IA-32 Architectures Optimization Reference Manual*, Intel Corporation, Aug. 2010.

[32] G. Lathoud, M. Magimai-Doss, B. Mesot, and H. Bourlard, "Unsupervised Spectral Subtraction for Noise-Robust ASR," in *Proc. of the IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, 2005.

[33] O. Yilmaz and S. Rickard, "Blind Separation of Speech Mixtures by Time-Frequency Masking," *IEEE Trans. on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, Jul. 2004.

[34] A. Roy, M. Magimai-Doss, and S. Marcel, "Phoneme Recognition using Boosted Binary Features," in *Proc. of IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.



Anindya Roy received the B.E. in Electronics and Telecommunications Engineering from the Bengal Engineering and Science University, Shibpur, India in 2005, and the M.Tech. in Electronics and Electrical Communications Engineering from the Indian Institute of Technology, Kharagpur, India in 2007. Currently, he is pursuing a PhD in Electrical Engineering at the École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, and is a research assistant at the Idiap Research Institute, Martigny, Switzerland. His research interests include speaker verification, automatic speech recognition, multimodal biometrics, speech processing and computer vision.

Mathew Magimai.-Doss (S '03, M'05) received the B.E. in Instrumentation and Control Engineering from the University of Madras, India in 1996; the M.S. by Research in Computer Science and Engineering from the Indian Institute of Technology, Madras, India in 1999; the PreDoctoral diploma and the Docteur ès Sciences (PhD) from École Polytechnique Fédérale de Lausanne (EPFL), Switzerland in 2000 and 2005, respectively. From April 2006 until March 2007, he was a postdoctoral fellow at International Computer Science Institute, Berkeley, USA. Since April 2007, he has been working as a research scientist at Idiap Research Institute, Martigny, Switzerland. His research interests include speech processing, automatic speech and speaker recognition, statistical pattern recognition and artificial neural networks.

Sébastien Marcel is senior research scientist at the Idiap Research Institute. He is interested in multimodal biometric person recognition, man-machine interaction and content-based multimedia indexing and retrieval. He has obtained his PhD in signal processing from Université de Rennes I in France (2000) at CNET, the research center of France Telecom (now Orange Labs). He currently leads the Biometric Person Recognition research team at the Idiap Research Institute and manages collaborative (CH and EU FP7) research projects. In 2010, he was appointed Visiting Professor at the University of Cagliari (IT) where he taught a series of lectures on "face recognition". He has served as a regular reviewer for a number of international journals and conferences. He is also a member of IEEE.