Speaker recognition using PCA-based feature transformation
Ahmed, Ahmed Isam; Chiverton, John P.; Ndzi, David L.; Becerra, Victor M.

Published in:
Speech Communication

DOI:
10.1016/j.specom.2019.04.001

Published: 31/07/2019

Document Version
Early version, also known as pre-print

Link to publication on the UWS Academic Portal

Citation for published version (APA):

Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact pure@uws.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Speaker recognition using PCA-based feature transformation

Ahmed Isam Ahmeda,*, John Chivertona, David Ndzb, Victor Becerraa

aSchool of Energy and Electronic Engineering, University of Portsmouth, Portsmouth, UK, PO1 3DJ
bSchool of Computing, Engineering and Physical Sciences, University of the West of Scotland, Paisley, UK, PA1 2BE

Abstract

This paper introduces a Weighted-Correlation Principal Component Analysis (WCR-PCA) for efficient transformation of speech features in speaker recognition. A Recurrent Neural Network (RNN) technique is also introduced to perform the weighted PCA. The weights are taken as the log-likelihood values from a Single Gaussian-Background Model (SG-BM). For speech features, we show that there are large differences between feature variances which makes covariance based PCA less optimal. A comparative study of the performance of speaker recognition is presented using weighted and unweighted correlation and covariance based PCA. Extensions to improve the extraction of MFCC and LPCC features of speech are also proposed. These are Odd Even banks MFCC (OE-MFCC) and Multitaper-Fitted LPCC. The methodologies are evaluated for the i-vector speaker recognition system. A subset of the 2010 NIST speaker recognition evaluation set is used in the performance testing in addition to evaluations on the VoxCeleb1 dataset. A relative improvement of 44% in terms of EER is found in the system performance using the NIST data and 18% using the VoxCeleb1 dataset.

Keywords: weighted principal component analysis, feature fusion, i-vector system.

1. Introduction

The interest in speaker recognition is currently rapidly growing. Speaker recognition is becoming widely used for different applications, e.g. access control (security), audio indexing and forensic applications (Beigi, 2011). The front-end of a speaker recognition system is important because it can greatly affect the overall system performance. Acoustic feature extraction can be considered to be the main task in the front-end of a speaker recognition system. The aim of this work is to improve the speaker recognition front-end to help increase the accuracy of the system. This is achieved by enhancing the extraction of Mel-Frequency Cepstral Coefficients (MFCC) and Linear Predictive Cepstral Coefficients; and

Declarations of interest: none
*Corresponding author

Email addresses: ahmed.ahmed@port.ac.uk (Ahmed Isam Ahmed), john.chiverton@port.ac.uk (John Chiverton), david.ndzi@uws.ac.uk (David Ndzi), victor.becerra@port.ac.uk (Victor Becerra)

Preprint submitted to Speech Communication March 6, 2019
second by introducing an efficient PCA-based feature transformation which includes single feature transformation and multiple feature fusion. Fusion of multiple features is where different aspects of information about the speech signal are pooled together. It is also sometimes viewed as feature extraction and is known to improve the performance of speaker recognition systems (Neustein and Patil, 2012).

Most speaker recognition systems use MFCC as speech features, see e.g. Tirumala et al. (2017). MFCC are based on psychoacoustic theory, see e.g. Davis and Mermelstein (1980). The popular i-vector speaker recognition system introduced by Dehak et al. (2011) depended mostly on MFCC as the source of speech features. MFCC feature extraction uses spectral-decomposition that mimics human auditory perception. Filter banks are used with more emphasis on lower frequencies, similar to the human auditory system. The log-energies of the filter bank outputs are correlated and the Discrete Cosine Transform (DCT) is used to decorrelate them, which produces the MFCC features.

In traditional MFCC, the DCT is applied to all of the filter log-energies. Narrow-band noise affects the entire set of DCT coefficients as every filter’s log-energy contributes to all the coefficients, see e.g. Sahidullah and Saha (2012). A block based MFCC was proposed for the extraction of the cepstral coefficients. The blocks prevent the three peaks associated with the formants of speech from affecting each other. In (Damper and Higgins, 2003), subbands were used in separate recognition systems with score fusion to tackle the problem of narrow band noise. Another work, Besacier and Bonastre (2000) addressed distortions caused by noisy environments that may partially affect the speech spectrum. This latter work also used score fusion of multiple recognition systems but with a different number of filters.

We focus on the selection of particular subsets of a filter bank in light of their associated covariance matrices. A filter bank based spectral-decomposition is a transformation of the speech spectrum. The performance of this transformation could be considered in terms of the residual correlation in the correlation matrix of the filter output. This is related to the theory behind MFCC. We therefore propose to extract cepstral coefficients from particular subsets of filter banks. The aim of this is to reduce the residual correlation of the covariance matrices or to at least prevent it from becoming higher than that of the full set.

Linear Predictive Cepstral Coefficient (LPCC) features are based on Linear Prediction Coding (LPC). LPCC models the speech production mechanisms. This makes LPCC a good feature candidate for combination with MFCC; as it adds knowledge from a different perspective. Filter banks can be seen as a type of filter bank quantization of the speech spectrum. Linear prediction analysis, on the other hand, is an adaptive quantization analysis in which the filter poles are distributed on the peaks of the spectrum (Dautrich et al., 1983). This can make linear prediction coefficients sensitive to spectral bias as well as spectral smoothness. We thus propose an LPCC extraction extension: based on multitaper (multi-window) spectrum estimation. This method has been found to improve MFCC features with the Gaussian Mixture Model-Universal Background Model (GMM-UBM) speaker recognition system by Kinnunen et al. (2010) and with the i-vector system (Alam et al., 2013). In multitaper spectrum estimation, the speech frame is windowed by multiple windows instead of the Hamming window and the outputs are averaged resulting in smooth spectral estimates.
Feature fusion is commonly based on a concatenation of features. Unfortunately, this increases the dimensionality. Zeinali et al. (2017) proposed an i-vector extractor for text-dependent speaker verification which concatenated MFCC and bottleneck features. Bottleneck features are extracted from acoustic features using a Deep Neural Network (DNN), see e.g. Yu and Deng (2014). MFCC and LPCC features were also concatenated to improve speaker identification in (Omar and El-Hawary, 2017). A method to combine feature selection and feature fusion based on multiple kernel learning was proposed for speaker emotion recognition by Jin et al. (2014). Concatenation of acoustic features of multiple channels was also found to enhance speech recognition by e.g. Tu et al. (2017).

Feature fusion through concatenation can improve performance. However, as already noted, the increased dimensionality is a negative by-product of this process which can reduce the overall impact of the resulting performance improvement. The increase in dimensionality also causes the required amount of data to increase exponentially for reliable density estimates of the GMM (Kinnunen and Li, 2010). Dimensionality reduction techniques, such as Principal Component Analysis (PCA), can thus be used to help overcome this shortcoming. PCA can be applied to each speech sample during training and or testing. However this adds another level of computations to an already relatively complex system such as the i-vector speaker recognition system. Furthermore, this will result in having speech features in different spaces. This can be appropriate for individual speaker modelling (Kwok et al., 2004) but not for other speech processing applications.

Alternatively, it is possible to perform PCA once to define one set of universal principal components. All speech samples’ features can then be projected onto a single, unified and reduced dimensional space. Another advantage of PCA is that the resulting principal components are orthogonal. This means that the projected speech features can be considered uncorrelated. This is useful, e.g. as Gaussian Mixture Models (GMM)s can be fitted assuming diagonal covariance matrices (Rao and Koolagudi, 2013). This method of defining global principal components was proposed for dimensionality reduction in speaker identification using the GMM-UBM system in (Seo et al., 2009). It was also found to outperform concatenation of features by Sarkar et al. (2014). The authors combined cepstral features and phonetically discriminant features for speaker verification. A similar technique by Zhang et al. (2016) also used global covariance PCA for feature fusion in the i-vector system. Other examples can be found in the literature on the use of PCA for speech feature fusion such as (Chibelushi et al., 1997), (Lee and Narayanan, 2005) and (Xie and Guan, 2013).

In Section 4, we show that there are large differences between the variances of cepstral coefficients in MFCC and LPCC features. If a set of variables have different scales, the correlation matrix is often more appropriate. Otherwise, the features with the higher variances will dominate the few principal components, see e.g. Jolliffe (2002). These high variance features may not have superior importance over other features. Another important consideration is with regards to noise. Some of the variation in the feature vectors is going to be from noise. However PCA is not able to distinguish between sources of variance (Delchambre, 2014; Bailey, 2012).

PCA can be performed using e.g. Singular Value Decomposition (SVD). However it has been found by Roweis (1998) that iterative PCA methods can identify the dominant
eigenvectors more correctly and efficiently. Roweis (1998) used Expectation-Maximisation (EM). A number of extensions of the EM method were introduced for PCA in the case of noisy or missing data by Bailey (2012). Then Delchambre (2014) proposed a power iteration method as an improvement over the EM algorithm of Bailey (2012). It was found to be superior in finding the principal components in the order of variance they represent. However, the power iteration method may suffer from a low convergence rate under particular conditions (Delchambre, 2014).

To tackle these aspects, we propose a weighted correlation based PCA. An iterative process based on a Recurrent Neural Network (RNN) is used. The weights are the log-likelihood values of a single Gaussian background model. This method is found to give the same principal components as the power iteration method, but it has a higher rate of convergence. One of the earliest work in using neural networks for PCA can be found in (Oja, 1982). A class of unconstrained Hebbian-type learning rules were derived and the dominant eigenvector was directly estimated from the input sequence not from a correlation or covariance matrix. Hence, it was not possible to include a weighted correlation or covariance matrix and the extraction of further eigenvectors was also not possible. The RNN method presented here is an extension of the work by Rajasekaran and Pai (2002) and Yi et al. (2004). There, the RNN was used to find the largest eigenvalue and the associated eigenvector of real symmetric matrices.

The proposed methodology is evaluated here using the i-vector speaker recognition system. A challenge was faced in the development of the i-vector speaker recognition system due to the lack of appropriate development data. We therefore introduce a data augmentation method to overcome this problem. The method is based on the idea of adding a simulated channel effect to increase the amount of development data. Previously, Garcia-Romero et al. (2012) added environmental noise to development speech signals. This was used to match similar noise embedded in the test speech samples for an i-vector system. Mak et al. (2016) trained a mixture of PLDA models and the presented test speech was directed to the PLDA model that best matched the test sample’s signal-to-noise ratio. In contrast to this, our proposed method aims to enable proper channel variability modelling. We use sample adaptive noise power. The evaluation set is potentially less overfit to the selection of the noise added to the development data.

Snyder et al. (2018) also used data augmentation to increase the amount of data. This was to improve the performance of a Deep Neural Network (DNN) based x-vector extraction system. These x-vectors are DNN embeddings which convert variable length speech samples into fixed length vectors, similar to i-vectors. Noise was randomly added by Snyder et al. (2018) to produce condition-variable samples which increased the amount of data for training. The effect of the data augmentation on an i-vector based verification framework was also investigated. However, the proposed methodology was more helpful in the x-vector based system.

The GMM-UBM/i-vector framework is adopted in this work. This is because the GMM-UBM can be estimated with the available data to sufficiently represent a global acoustic space. This also makes it fast to re-estimate which is useful; the evaluation of the system performance with PCA requires the re-development of the i-vector system many times. Ac-
cording to the most recent performance reported by Khosravani and Homayounpour (2018) using the same evaluation set, the DNN/i-vector framework (Lei et al., 2014) had a lower EER of 0.28% than the GMM-UBM framework.

The proposed data augmentation method enables a reasonable level of system performance. This is in terms of the Det5 condition of the 2010 NIST evaluation dataset using MFCC features. Note that data augmentation is only designed to compensate for lack of development data. Therefore it is not used with enrolment or test samples. The paper is organised as follows. The development of the i-vector system using data augmentation is described in Section 2. The extraction of OE-MFCC and multitaper-fitted LPCC features are reported in Section 3. The description of principal component analysis is presented in Section 4. Results of the proposed methodology are reported in Section 5 followed by conclusions in Section 6.

2. Data Augmentation for the Establishment of The i-vector System

The i-vector speaker recognition system models inter-speaker and intra-speaker variability (total variability) simultaneously according to Dehak et al. (2011). The goal of intra-speaker variability (session/channel variability) modelling is to reduce the effect of channel mismatch. This can happen between enrolment and test speech in speaker recognition. Thus, the establishment of the system requires speech samples recorded over different channels. This data is also used to perform Linear Discriminant Analysis (LDA) and Probabilistic Linear Discriminant Analysis (PLDA). If the development data is not sufficient, the i-vector system cannot perform appropriately. It would not be able to model the session variability for speakers (intra-speaker variability). Such development data is not widely available to researchers hence a data augmentation technique is introduced here to tackle the problem. The goal of this technique is to use one speech recording for a speaker to produce a channel-variable recording by adding a simulated channel effect to the available recording.

According to the simple factor analysis model of the i-vector (Dehak et al., 2011), a speech sample is expressed as

\[ m_u = m + Tw, \]

where \( m_u \) is a speaker-and-channel dependent supervector, \( m \) is the speaker-and-channel independent supervector, \( T \) is the total variability subspace and \( w \) is the i-vector. Our data augmentation is based on a theoretical speaker synthesis model presented in (Teunen et al., 2000). The model tackles the problem of channel mismatch between enrolment and test samples. Two (enrolment and test) utterances of the same speaker with ‘speaker and channel’-dependent supervectors \( m_u \) and \( \tilde{m}_u \) can be considered. The model uses the assumption that \( \tilde{m}_u \) is synthesised from \( m_u \) by adding a supervector \( c \). This depends on the channel conditions of the two utterances (Kenny et al., 2007), with

\[ \tilde{m}_u = m_u + c, \]

where \( c \) is assumed to be a channel compensation supervector with normal distribution. We fl that assumption here by passing the speech signal through a Gaussian channel in order
to incur a different channel effect on the available recordings. Hence, any available recording becomes two recordings: the original one and the one with the added Gaussian noise. Since most of the available utterances are longer than 2 minutes, the utterances are split into two to further improve the available number of samples. This is based on the observation that system performance is more influenced (degraded) by utterance lengths of less than 1 minute, (Rao and Mak, 2013).

The reason for using a Gaussian channel is that Gaussian noise is used to mimic random processes that occur in nature (Houdré et al., 2016). It is also used to model many practical channels like wired and wireless telephone channels. The additive noise in such channels are due to a combination of causes. By the central limit theorem, the cumulative effect of a number of random effects will be approximately normal thus the Gaussian assumption becomes valid, see e.g. (Cover and Thomas, 2012).

The power of the noise added here is controlled such that the produced speech signals maintain a sufficient signal-to-noise (SNR) ratio denoted by $\eta_d$. In other words, the signal power is taken into account to prevent the added noise power from becoming destructive. The added Gaussian noise is defined by a normally distributed random vector $\mathbf{r} = [r_1, r_2, ..., r_N]$. The elements of this vector $r_n$ are defined by a common mean of zero and variance $\phi_r$, i.e. $r_n \sim N(0, \phi_r)$. Since this Gaussian noise, represented by $\mathbf{r}$, has a mean of zero, then its power is equal to its variance $\phi_r$.

The value of $\eta_d$ is empirically selected based on the performance of the system indicated by the Equal Error Rate (EER). The EER value indicates the operating point where the system’s false acceptance rate is equal to its false rejection rate (Beigi, 2011). $\eta_d$ of 30dB is found to present the best performance with the lowest EER. The performance with other values of $\eta_d$ have also been investigated. At $\eta_d = 20$dB the performance is found to degrade, compared to that provided at $\eta_d = 30$dB. The utterances with added Gaussian noise become very noisy which appears to be harmful to the system. The performance is also found to degrade at $\eta_d = 40$dB. There, the Gaussian noise power is very low, thus the utterances with added noise are not very different from the original utterances.

3. Acoustic Feature Extraction

In this section, we introduce feature extraction improvements in OE-MFCC and multitaper-LPCC features.

3.1. Odd-Even Filter Banks MFCC (OE-MFCC) Extraction

Conventional Mel Filter Banks (FB)s comprise of filters that are overlapped (by 50%). This is in order not to lose the parts of the speech spectrum where it is attenuated by the edges of each filter. Due to this overlap, the log energy of a particular filter somewhat resembles that of the adjacent ones especially if they (all three) capture a slowly varying section of the spectrum. Hence, a set of overlapping filter may result in relatively high residual correlation in the covariance matrix of the filter banks’ log-energies. We believe that avoiding the overlap between the filter banks can reduce the residual correlation of the covariance matrix. As this can cause spectrum loss, we propose to separate the odd indexed
and even indexed fi of a fi bank, illustrated in Fig. 1a. Then to extract cepstral coefficients separately from each subset.

The proposed methodology can have the following advantages: it decreases the residual correlation for each subset (as assumed to be desired); no spectrum is lost compared to an overlapped fi r bank; the effect of narrow-band noise on the cepstral coefficients is reduced; and the computation complexity in performing the DCT is minimised. It can also compensate for the limitation of extracting higher order cepstral coefficients in standard MFCC. It should be noted that higher order coefficients of MFCC are more susceptible to noise (Reyes-Galaviz and García, 2009).

![Figure 1](image)

(a) Odd even filters subsets  (b) Residual correlation

Figure 1: This figure illustrates odd and even subsets of a full set of overlapping filters. It also illustrates the residual correlation of the filter bank function for different values of the correlation coefficient of a Markov-1 process covariance matrix.

Calculating the energy of odd and even indexed fi separately has been used before to achieve computational efficiency in a hardware implementation of MFCC by Jo et al. (2016). Vu et al. (2010) used only the odd fi and the points of the even fi rs were determined by subtracting each odd fi from 1. However, in both of these examples, all odd and even fi log-energies were pooled together and then DCT was applied.

The residual correlation is the mean of the absolute values of the off-diagonal elements of a correlation matrix (Sahidullah and Saha, 2012). The residual correlation (E) of the fi bank function (overlapped and non-overlapped) is evaluated for a fi order Markov process covariance matrix with different correlation coefficients (ρ). More on this procedure can be found in (Poularikas, 2010). Fig. 1b shows that the residual correlation of overlapped fi is higher than any of the odd-indexed or even-indexed subsets. It can also be noticed that the diff increases for higher values of ρ.

Table 1 reports the residual correlation in the correlation matrix of the fi banks’ log-energies for speech data. The speech data is the training samples of the 2002 NIST SRE dataset (139 males and 139 females). It can be seen that for three cases of fi er banks, the odd and even subsets exhibit lower residual correlation than that of the full set.

Cepstral coefficients extracted from both odd and even subsets are concatenated here for use in the speaker recognition system. Both subsets interchangeably cover the full band of the speech spectrum. This relatively increases the residual correlation in the correlation
Table 1: Residual correlation of the filter banks log-energies correlation matrix.

<table>
<thead>
<tr>
<th>No. of Filters</th>
<th>Full Set (E)</th>
<th>Odd Subset (E)</th>
<th>Even Subset (E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.6547</td>
<td>0.6339</td>
<td>0.6416</td>
</tr>
<tr>
<td>24</td>
<td>0.6486</td>
<td>0.6314</td>
<td>0.6367</td>
</tr>
<tr>
<td>28</td>
<td>0.6415</td>
<td>0.6258</td>
<td>0.6321</td>
</tr>
</tbody>
</table>

matrix of their cepstral coefficients. However, the performance of speaker recognition does not appear to be particularly sensitive to it. In block MFCC of (Sahidullah and Saha, 2012), the cepstral coefficients of overlapping blocks of fi exhibited relatively higher residual correlation in their correlation matrix. They, however, generally presented better performance than some of the other forms of block MFCC presented.

As well as the analysis of the residual correlation, a further experiment is conducted here to assess the likeliness of peer (from an order perspective) cepstral coefficients of odd and even subsets. The experiment shows that, except for a few, there exists relatively high diversity between peer cepstral coefficients of the odd and even subsets. In the experiment, for a particular speech utterance, cepstral coefficients are extracted with two subsets of 14 odd and 14 even fi of a fi bank. Let \( \mathbf{M}_1 \) and \( \mathbf{M}_2 \) represent the cepstral coefficients of the odd and even subsets, respectively. For the sake of comparison, 13 cepstral coefficients are extracted for the same utterance with a set of 14 overlapping fi fi bank. Denote this set of cepstral coefficients by \( \mathbf{M} \).

Afterwards, we measured the degree of correlation between the sets of cepstral coefficients using (see e.g. Sharma (2005))

\[
\rho(\mathbf{M}_1(o), \mathbf{M}_2(o)) = \frac{\mathbf{M}_1(o)\mathbf{M}_2(o)^T}{\|\mathbf{M}_1(o)\|_2 \|\mathbf{M}_2(o)\|_2},
\]

where \( \rho(\mathbf{M}_1(o), \mathbf{M}_2(o)) \) is Pearson’s correlation coefficient between the \( o \)th order cepstral coefficients of set \( \mathbf{M}_1 \) and set \( \mathbf{M}_2 \). The correlation coefficients, \( \rho(\mathbf{M}(o), \mathbf{M}_1(o)) \) and \( \rho(\mathbf{M}(o), \mathbf{M}_2(o)) \) are also calculated using (3). Note that the means of \( \mathbf{M} \), \( \mathbf{M}_1 \) and \( \mathbf{M}_2 \) must be normalised in order to calculate Pearson’s correlation coefficient.

Fig. 2 shows the correlation between all of these sets of cepstral coefficients for 13 orders of DCT coefficients. Compared to \( \rho(\mathbf{M}(o), \mathbf{M}_1(o)) \) and \( \rho(\mathbf{M}(o), \mathbf{M}_2(o)) \) cases, one can notice that there is low (peer) correlation between the cepstral coefficients of the odd and even subsets as indicated by \( \rho(\mathbf{M}_1(o), \mathbf{M}_2(o)) \). These fi can be used as an indicator to remove cepstral coefficients from OE-MFCC if they consist of redundant information which may harm the system performance.

As will be seen shortly, this new OE-MFCC feature is found to improve speaker recognition performance. This is a new paradigm for fi bank analysis compared to standard MFCC. We investigate the performance in relation to the number of fi rs and cepstral coefficients. This is reported in the results section.
3.2. Multitaper-Fitted LPCC

Linear prediction models speech production as an autoregressive process. This is where a speech frame can be predicted from past frames (delayed versions of the frame). This process is fitted to an all-pole digital filter model where the coefficients of the filter represent the vocal tract (the spectral envelop). Hence, the goal is to fit the filter coefficients that minimise the error between the speech frame and its predicted version. This in turn is realised using autocorrelation (Broersen, 2006). LPCC are calculated using a recursive process (Beigi, 2011).

Spectrum bias is mostly caused by spectral leakage and can be reduced by using a window function. Commonly, a Hamming window (Neustein and Patil, 2012) is used. Nevertheless, a single window is not an optimal choice as it down-weights the speech frame values at the edges of the window causing loss of information. Spectral leakage is also not minimised. Hence the chance of spectral bias persists (Prieto et al., 2007). Multitaper spectrum estimation was first presented in (Thomson, 1982). The multiple tapers are weighted. The resulting spectrum is smoothed, has less variance and the spectral leakage is minimised giving a reduced bias (Prieto et al., 2007). The estimated multitaper power spectrum is a weighted sum of these tapers given by

\[ \hat{s}[k] = \sum_{m=1}^{M} w_m \lambda_m[n] s[n] \exp \left( -j2\pi \frac{nk}{N} \right)^2, \]

where \( M \) is the number of tapers, \( \lambda_m \) is a taper associated with a weight \( w_m \), \( s[n] \) is a speech sample and \( N \) is the number of speech samples.

The determination of the LPC coefficients starts by framing the speech signal and then a window function (Hamming) is applied. They are then passed to the autocorrelation process. One of the methods to calculate the autocorrelation function is based on the Wiener-Khinchin theorem. This states that the Fourier transform of the autocorrelation function is equal to the power spectrum. Hence, the inverse Fourier transform of the power spectrum is the autocorrelation function (Kantz and Schreiber, 2004).
To incorporate the multitaper method, we determine the autocorrelation function by computing the inverse Fourier transform of the multitaper power spectrum,

\[ \hat{r}_{ss}[n] = \frac{1}{K} \sum_{k=1}^{K-1} \hat{s}[k] \exp\left(j2\pi \frac{nk}{K}\right). \]  

(5)

The multitaper type and the number of tapers are determined empirically based on the speaker recognition performance. This is discussed in the results section.

4. Weighted Principal Component Analysis

The principal components are commonly considered to be the eigenvectors of the covariance matrix. The associated eigenvalues are the amount of variance for the data in the direction of the respective eigenvectors. Let \( \mathbf{X} \) represent a matrix of feature vectors pooled together from a population of speakers. In the context of this paper, \( \mathbf{X} \) may represent MFCC, OE-MFCC, MFCC+LPCC or OE-MFCC+LPCC \(^1\) features. To perform PCA, the matrix \( \mathbf{X} \) must be mean normalised. The covariance matrix of\( \mathbf{X} \) can then be expressed as

\[ \Sigma = \mathbf{X}\mathbf{X}^T. \]  

(6)

If \( \mathbf{X} \) is also variance normalised then \( \Sigma \) of (6) becomes the correlation matrix, denoted here by \( \hat{\Sigma} \). Speech features can have high variability between their variances. This suggests that the eigenvectors of the correlation matrix of \( \mathbf{X} \) should be the ones taken to be the principal components instead of the covariance matrix of \( \mathbf{X} \).

A weighted correlation or covariance matrix can be determined by including a weights matrix \( \mathbf{W} \). It is the same size as the feature vector matrix \( \mathbf{X} \). Similarly, if only the mean of \( \mathbf{X} \) is normalised, then its weighted covariance matrix can be achieved as in the following

\[ \Sigma_w = (\mathbf{X} \circ \mathbf{W})(\mathbf{X} \circ \mathbf{W})^T \odot (\mathbf{W}\mathbf{W}^T), \]  

(7)

where \( \circ \) and \( \odot \) indicate the Hadamard product and division, respectively. The weighted correlation matrix \( \hat{\Sigma}_w \) can also be determined using (7) if the variance of \( \mathbf{X} \) is normalised.

Fig. 3 shows the variances of the cepstral coefficients of MFCC and LPCC features. It can be seen that there is large variation in the variances of the cepstral coefficients. The first and second derivatives of the cepstral coefficients shown in the figures have even smaller values. The MFCC lower order coefficients are known to be more sensitive to undesirable effects caused by factors like the transmission channel (Tan and Lindberg, 2008), yet they have relatively high variance. According to (Jolliffe, 2002), these coefficients will dominate the few principal components if the covariance matrix is used for PCA.

The logarithms of the elements of the covariance and correlation matrices of \( \mathbf{X} \) are depicted in Fig. 4. Here the feature vectors of \( \mathbf{X} \) are 13 MFCC coefficients appended with their first and second derivatives. It can be seen that neighbouring covariance matrix elements

\(^1\)The symbol, +, indicates a concatenation of the features prior to PCA.
Figure 3: Variances of MFCC and LPCC cepstral coefficients.

Figure 4: Left image: covariance matrix. Right image: correlation matrix. Note logarithmic scale used and tiled pattern due to derivatives being included in the feature vectors.
appear to have relatively small differences in variance values of the cepstral coefficients. For example, in the top left corner the highest attributes are associated with the higher variance coefficients. On the other hand, the relationship between the cepstral coefficients expressed by the correlation matrix, do not seem to be affected by those numerical variations. For example, the diagonal of the correlation matrix contains the highest values in the matrix. These are the correlation of each cepstral coefficient with itself. Hence, it appears to be very important for speech features that the feature variances are normalised before performing any analysis for PCA.

4.1. Recurrent Neural Network

This section introduces a class of Recurrent Neural Networks (RNN)s that can be used to extract the principal components of weighted covariance and correlation matrices. The type of the RNN can be defined by its architecture. Furthermore an RNN can be designed to model a dynamical system. Rajasekaran and Pai (2002) formulated the eigendecomposition problem as an equilibrium problem for a dynamical model of a RNN. In that work, the RNN was used to identify the largest eigenvalue and the associated eigenvector of a real symmetric matrix. Similarly in (Yi et al., 2004), a class of RNN was proposed to determine the largest and smallest eigenvalues and the associated eigenvectors.

The RNN presented in (Rajasekaran and Pai, 2002) had finite weights which were the elements of a real symmetric matrix. When the network input was made to be an arbitrary vector, the output converged to the equilibrium state. This was shown to be the dominant eigenvector. This work defined the objective of the learning algorithm of (Rajasekaran and Pai, 2002) and considers it for the eigendecomposition problem presented here. The RNN described here is capable of identifying the desired subset of the weighted principal components. The principal components are extracted in the order of the size of the eigenvalues from the weighted covariance or correlation matrix. This is not possible with conventional SVD because the SVD solution does not use or calculate a correlation or covariance matrix. Instead SVD identifies the principal components directly from a sequence of feature vectors. Thus, it can be difficult to engage any weighting.

Weighted covariance and correlation matrices of a set of speech feature vectors are real and symmetric. The methodology described here considers the weighted correlation matrix \( \tilde{\Sigma}_w \) and can be equally applied to the weighted covariance matrix \( \Sigma_w \). The size of \( \tilde{\Sigma}_w \) is \( D \times D \), where \( D \) is the feature dimensionality. Consider the following eigendecomposition formula

\[
\tilde{\Sigma}_w \mathbf{p}_w = \gamma_w \mathbf{p}_w,
\]

where \( \mathbf{p}_w \) is the weighted principal component associated with eigenvalue \( \gamma_w \). The learning algorithm for determining \( \mathbf{p}_w \) using the RNN is now described.

The structure of the RNN requires two layers: a variable layer and a constraint layer. The number of nodes in each layer is equal to \( D \). Both layers are fully interconnected with the weights being the elements of the weighted correlation matrix \( \tilde{\Sigma}_w \). The initial input to the neurons of the variable layer can be the values of a random column vector \( \mathbf{v}^{(1)} \). Let
\( \mathbf{C} = \tilde{\Sigma}_w \), the output of the neurons of the constraint layer at iteration \( t \) will be

\[
\mathbf{v}^{(t)}(j) = \sum_{i=1}^{D} \mathbf{C}_{ij} \mathbf{v}^{(t)}_i \quad \text{for} \quad j = 1, 2, ..., D,
\]

(9)

where \( i, j = 1, 2, ..., D \) are, respectively, the rows and columns of \( \mathbf{C} \). This describes the so-called feed-forward step. Now let \( \mathbf{y}^{(t)} \) be a column vector of the values of \( \mathbf{v}^{(t)}(j) \) arranged from \( j = 1 \) to \( D \). In the feedback step, the neural links from the constraint layer to the variable layer are \( 1/\hat{\gamma}^{(t)} \), where

\[
\hat{\gamma}^{(t)} = \max \mathbf{y}^{(t)},
\]

(10)

which can also be considered to be the eigenvalue at iteration \( t \). For the next iteration, the input to the variable layer will be

\[
\mathbf{v}^{(t+1)} = \frac{\mathbf{y}^{(t)}}{\hat{\gamma}^{(t)}}.
\]

(11)

The process described by (9), (10) and (11) is repeated for \( \kappa \) iterations until the network converges to the equilibrium state with the most dominant eigenvector,

\[
\mathbf{v}^{(\kappa)} = \frac{\mathbf{y}^{(\kappa-1)}}{\hat{\gamma}^{(\kappa-1)}}.
\]

(12)

Using the variables of the learning algorithm, one can re-write (8) as

\[
\mathbf{C} \mathbf{v}^{(\kappa)} = \hat{\gamma}^{(\kappa)} \mathbf{v}^{(\kappa)}.
\]

(13)

The outcome of the operations on both sides of (13) is a column vector. The objective of the learning algorithm here is to minimise a parameter \( \alpha \) defined by

\[
\alpha = \mathbf{C} \mathbf{v}^{(\kappa)} - \hat{\gamma}^{(\kappa)} \mathbf{v}^{(\kappa)} | | \mathbf{v}^{(\kappa)} | |.
\]

(14)

In order to meet the learning objective, i.e. making the value of \( \alpha \) approach zero, the learning algorithm must be sufficiently iterated. This is shortly demonstrated below.

It can be seen from (8) that the real symmetric matrix, \( \tilde{\Sigma}_w \), scales the eigenvector, \( \mathbf{p}_w \), by the eigenvalue, \( \gamma_w \). If the largest element of \( \mathbf{p}_w \) is equal to one then the largest element of \( \gamma_w \mathbf{p}_w \) is equal to the eigenvalue \( \gamma_w \). One can notice that the calculation in (9) estimates the right hand side of (8), \( \gamma_w \mathbf{p}_w \), given the parameters of the left hand side, \( \tilde{\Sigma}_w \) and \( \mathbf{p}_w \). By comparing (10) and (11), one can infer that the maximum value of any \( \mathbf{v}^{(t)} \), for \( t > 1 \), is equal to one. This justifies the calculation of \( \hat{\gamma}^{(t)} \) using (10) since \( \mathbf{y}^{(t)} \) is equivalent to \( \hat{\gamma}^{(t)} \mathbf{v}^{(t)} \).

The dominant weighted principal component, \( \mathbf{p}_w \), of \( \tilde{\Sigma}_w \) is given here by normalising the dominant eigenvector, \( \mathbf{v}^{(\kappa)} \), to unity

\[
\mathbf{p}_w = \frac{\mathbf{v}^{(\kappa)}}{| | \mathbf{v}^{(\kappa)} | |}.
\]

(15)
and the associated eigenvalue is now calculated using $\mathbf{p}_w$ and $\tilde{\Sigma}_w$, as follows

$$\gamma_w = \mathbf{p}_w^T \tilde{\Sigma}_w \mathbf{p}_w.$$  \hfill (16)

The rest of the weighted principal components are determined as follows. The variance captured by the first principal component, $\mathbf{p}_w$, is removed from $\tilde{\Sigma}_w$ as in the following

$$\tilde{\Sigma}_w = \tilde{\Sigma}_w - \mathbf{p}_w \gamma_w \mathbf{p}_w^T,$$ \hfill (17)

then the same pre-described learning algorithm can be applied using $\tilde{\Sigma}_w$ as the network weights to obtain the second principal component. This procedure is repeated as many times as required to obtain the desired set of $d$, where $d \leq D$, weighted principal components.

Fig. 5 shows that the proposed RNN solution for the eigendecomposition problem meets the objective of the learning algorithm. One can observe that $\alpha$ approaches zero with a sufficient number of iterations. Fig. 5 demonstrates the case when the network input is an arbitrary vector. The use of such an arbitrary vector may not be optimal. This was previously discussed in (Delchambre, 2014) for the power iteration method. It was suggested that if some prior eigenvectors were available, it would then be better to use those to start the iterative process because of the relevance to the problem.

In this work, we propose that for every weighted principal component to be extracted, the iterative process is started with the corresponding unweighted principal component determined using SVD. The RNN solution can then be viewed as a process of updating the SVD principal component using the weighted correlation matrix. This strategy can at least increase the convergence rate as discussed in (Delchambre, 2014).

![Figure 5: Demonstration of how the learning objective, minimising $\alpha$, of the proposed RNN solution is being met. Examples of the extraction of the first and second dominant principal components.](image)

**4.2. Weighting Criterion**

Each feature vector of the data used to extract the principal components is assigned a weight. This is important because we want to decrease the significance of those feature vectors that are noisy or that represent silence. The proposed weighting criterion can be described as follows. Using the EM algorithm (Reynolds and Rose, 1995), a GMM is fit to
the feature vectors that are used in the PCA. Then the corresponding log-likelihood values of the GMM are used as weights. This criterion is motivated by the concept of acoustic space modelling for speech with the GMM-UBM (Reynolds et al., 2000). Also, by the methods of model based Voice Activity Detection (VAD) as in (Anguera et al., 2006). It is therefore anticipated that bad feature vectors will have relatively low log-likelihood values thus lower weights.

![Figure 6: Weight variability for the case of SG-BM versus the case of the GMM-UBM with different number of components.](image)

The GMM used here has one component and it is referred to as a single Gaussian background model (SG-BM). One might argue that a GMM-UBM can be used, however, it seems to overfit for our particular modelling approach. The reason is that higher variability between the weights of the feature vectors can be seen with the SG-BM as illustrated in Fig. 6. In the same figure we can see that by using a GMM-UBM, the variability between the weights is less. It also decreases as the number of the mixture components increases.

5. Experiments and Results

In this section the results of the aforementioned techniques are now presented. The performance of the i-vector system for speaker recognition is reported in terms of the Equal Error Rate (EER). For convenience, a subsection is allocated for each part of the work. Feature transformation and fusion are achieved based on the proposed RNN approach for weighted correlation PCA. For the sake of comparison, weighted covariance PCA is also considered. Weighted correlation and weighted covariance PCA are referred to as WCR-PCA and WCV-PCA, respectively.

The RNN PCA method is found to give the same principal components as the power iteration method and exactly the same results for speaker recognition using the i-vector system. However, in light of some conditions regarding the power iteration method as addressed in (Delchambre, 2014), this method can be superior in guaranteeing the extraction of the dominant eigenvector(s). A detailed investigation is beyond the scope of this paper.
However, the RNN PCA method has a higher convergence rate compared to the power iteration method as depicted in Fig. 7.

![Figure 7: Comparison of the convergence rates of the power iteration method and the recurrent neural network method for the extraction of the first weighted principal component.](image)

Additionally, the performance introduced by using the classical SVD solution for PCA is also presented. SVD is used to decompose variance normalised features and non-variance normalised features which are equivalent to the eigendecomposition of the correlation and covariance matrices, respectively. The unweighted correlation PCA is referred to as CR-PCA and the unweighted covariance PCA is referred to as CV-PCA.

The data used for the PCA is the same as the one used to estimate the GMM-UBM. This will be described shortly, in Section 5.2. Note that the number of speakers of both genders is balanced. The number of iterations of the RNN which is used to extract the principal components of the WCR-PCA and WCV-PCA are 50.

Each feature and feature combination is normalised using the Cepstral Mean and Variance Normalisation (CMVN) over a sliding window of 3s worth of feature vectors. For covariance based PCA, features and feature combinations of the speech utterances are mean normalised and projected onto (multiplied by) the principal components. CMVN is then used to normalise the resultant features using a sliding window. For correlation based PCA, features and feature combinations of the speech utterances are subject to mean and variance normalisation and projected onto (multiplied by) the principal components. CMVN is also used to normalise the resultant features using a sliding window. One can notice that, before projecting the feature vectors onto the principal components, a different feature standardisation is followed for each case. This is necessary to accommodate the feature space in which the principal components were estimated.

For weighted PCA, the weights are only involved in the extraction of the weighted principal components. This is done by associating the weights in the calculation of the weighted correlation and covariance matrices using (7). The weights are calculated as described in Section 4.2 using the same feature vectors (and feature type) used to estimate the principal components. In Tables 6, 7, 8 and 9, \( \delta \) indicates the number of the principal components used for the projection of the original features to the new reduced dimension feature space.
‘All’ means all the principal components (also the original features dimension). ‘AV’ and ‘STD’ respectively refer to the average and standard deviation of EER. The amount of variance captured by the reported number of principal components is approximately in the range of 95% to 99% for CV-PCA/WCV-PCA and 85% to 95% for CR-PCA/WCR-PCA. These ranges of variances are found to give the best recognition performance in terms of average EER. The reduction in computation time as a result of reduced system complexity is presented in Section 5.7.

Subsections 5.1 to 5.7 include the main body of the experimental results, specifically from telephone speech data (for details see subsection 5.1). Subsection 5.8 presents additional evaluations on Youtube data.

5.1. Corpora

The data used for the development of the system includes the NIST 2002 SRE telephone training data (English) (Martin and Mark, 2004), the NCHLT Speech Recognition microphone corpus (English) (De Vries et al., 2014) and the LWAZI Speech Recognition telephone corpus (English, Afrikaans, Sesotho and Zulu) (Barnard et al., 2009). The system is gender-independent and in order to balance the analysis, the number of development speakers is 639 males and 639 females speakers (1278 speakers). Speech recording with an average length of 2 minutes can be obtained from these datasets for each development speaker. Table 2 summarises the number of utterances available, the number after splitting and after adding the simulated channel effect.

<table>
<thead>
<tr>
<th>No. of Available Utterances</th>
<th>Splitting</th>
<th>Adding Channel Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per speaker</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>For all Speakers</td>
<td>1278</td>
<td>2556</td>
</tr>
</tbody>
</table>

Table 2: Summary of the number of utterances for each speaker and for all the speakers.

The performance is evaluated on a subset of the 2010 NIST speaker recognition evaluation dataset (Martin and Greenberg, 2010). This subset is the core-core evaluation condition (commonly referred to as Det5) which contains telephone speech for enrolment and test data.

5.2. Performance of Data Augmentation in the Development of the i-vector System

This subsection investigates the presented data augmentation method. The performance of the i-vector system is investigated using limited data that is expanded using the data augmentation technique. A model-based voice activity detection is used to remove silences from the speech utterances (Sohn et al., 1999). For the development data, the silences are removed from the original utterances before adding the Gaussian noise. Acoustic features are extracted from speech frames of 25 ms length with 10 ms frame shift. For MFCC and OE-MFCC features, the power spectrum of the speech frames is estimated using the multitaper method with four multipeak tapers as in (Kinnunen et al., 2010). The total number of fi in the filter banks are 24 for MFCC and 28 for OE-MFCC (14 odd and 14 even). Then 13
cepstral coefficients (excluding the 0th coefficient) are obtained by applying DCT to each set and subset of fi banks’ log-energies. The cepstral coefficients are appended with their fi and second derivatives.

For OE-MFCC, it was experimentally found that the fi two coefficients of the 13 basic cepstral coefficients degrade the performance if they are kept together from both the odd and even subsets. This is possibly because keeping these two coefficients only result in redundant information as they are highly correlated between odd and even subsets (see fi 2). Hence, these two coefficients are removed for the even subset all through the experiments presented but their fi and second derivatives are kept.

For multitaper-fitted LPCC, the autocorrelation function is determined using the inverse of a multitaper power spectrum. The multitaper power spectrum is estimated with four multipeak tapers. 12 LPC coefficients are calculated and used to extract 13 LPCC coefficients appended with their fi and second derivatives for a total of 39 coefficients.

The original recordings of NIST 2002 SRE telephone training data (139 males and 139 females) are used to estimate a gender-independent GMM-UBM with 2048 mixtures. The rest of the system’s development steps use the augmented data (see Table 2). This includes: estimating the total variability matrix; performing the LDA analysis; and training the Gaussian PLDA model. The i-vectors are 400 in size. This is then reduced to 150 using LDA. Before training the Gaussian PLDA model, the i-vectors are centred, then they are subject to length normalisation and a whitening transformation. These parameters are fi for the system except for the features which will be constantly changing.

The proposed data augmentation based on simulated Gaussian channel effect is considered here. Utterance splitting presented in previous work (Rao and Mak, 2013) is also used here for the same purpose of increasing the amount of development data. In order to purely evaluate the effect of adding Gaussian noise, utterance splitting is fi deployed. A total of 2556 utterances are used to establish the system’s initial performance. Afterwards, Gaussian noise is used to increase the number of utterances to 5112 which is more appropriate for the development of the system. For just the MFCC features, Fig. 8 demonstrates the effect of adding Gaussian noise on system performance. Moreover, Table 3 reports more results on this effect for different features and feature combinations. The features listed in Table 3 are going to be used with PCA and the performance achieved with ‘splitting plus adding channel effect’ will be used as a reference for assessing the methodology presented for PCA.

<table>
<thead>
<tr>
<th>Features (Filter Bank Size)</th>
<th>Feature Dimensionality</th>
<th>EER(%) Splitting</th>
<th>EER(%) Splitting plus Adding Channel Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC (24)</td>
<td>39</td>
<td>8.20</td>
<td>3.79</td>
</tr>
<tr>
<td>OE-MFCC (28)</td>
<td>76</td>
<td>7.97</td>
<td>3.16</td>
</tr>
<tr>
<td>MFCC (24) + LPCC</td>
<td>39 + 39</td>
<td>8.01</td>
<td>3.60</td>
</tr>
<tr>
<td>OE-MFCC (28) + LPCC</td>
<td>76 + 39</td>
<td>8.33</td>
<td>3.76</td>
</tr>
</tbody>
</table>

Table 3: Effect of using Gaussian noise in data augmentation for different features and feature combinations.
5.3. System Performance using OE-MFCC Features

This subsection includes a study of speaker recognition with OE-MFCC. A number of parameter variations are considered. It also presents a comparison with conventional MFCC and block based MFCC. Fig. 9 demonstrates the system performance for OE-MFCC and MFCC. A comparison is made between a Hamming window and multitaper (four multipeak tapers) spectrum smoothing. Results are also a function of the number of \( f_i \) in the \( f_i \) banks. The feature dimension is not varied for this part, 39 and 76 for MFCC and OE-MFCC, respectively. It is notable that OE-MFCC greatly benefits from multitaper spectrum estimation. This has resulted in lower EER than MFCC for all numbers of \( f_i \). For the Hamming window case, OE-MFCC presents better performance in a number of cases and the lowest EER (compared to a Hamming window MFCC) for 35 \( f_i \). OE-MFCC also has a relatively stable low EER operation point in the range of 32 to 34 \( f_i \) (in the \( f_i \) bank).

For the spectrums estimated using the multitaper method, Fig. 10 illustrates the performance of OE-MFCC for a lower number of \( f_i \) (in the \( f_i \) bank(s)). This is also shown in comparison to MFCC. The dimensionality of the MFCC features is the number of \( f_i \) minus one plus delta (derivatives) and double delta (second derivatives). OE-MFCC feature dimensionality is the MFCC’s total number of features minus \( f_i \). For example, with a \( f_i \) bank of 28 \( f_i \) the MFCC dimensionality is 81; and so the OE-MFCC dimensionality is 76. This includes the derivatives, the second derivatives and then removing two coefficients from OE-MFCC as explained in section 5.2. Fig. 10 indicates for the majority of cases, the performance of OE-MFCC is superior to MFCC. This appears to be true even for fewer
Figure 9: The performance of OE-MFCC and MFCC in relation to the number of filter bank filters for fixed feature dimensionality. The feature dimensionality is 39 for MFCC and 76 for OE-MFCC (including delta and double delta coefficients).

Figure 10: System performance, OE-MFCC and MFCC, with a variable number of filters (in the filter bank(s)) and feature dimensionality.
fi in the fi bank(s). This is lower than those addressed in Fig. 9 and also whilst varying the feature dimensionality as well.

Tables 4 and 5 compare the performance of OE-MFCC, block MFCC and MFCC. The use of multitaper and Hamming window spectrum smoothing are investigated. Non-overlapping block based MFCC is referred to as Non-Overlapped Block Transformation (NOBT). Overlapping block based MFCC is referred to as Overlapped Block Transformation (OBT). In NOBT with two blocks, the result appears to have good performance (over MFCC) where the fi block covers the frequency band 0-883.17 Hz and the second block covers the band 745.93-4000 Hz. For a fi bank of 20 fi rs, the fi block includes the 1st to the 8th fi and the second block includes the 9th to the 20th fi. The frequency coverage of the blocks is accounted for here when the number of fi is higher than 20. For OBT, the blocks are allowed to overlap by one fi. A larger amount of overlap was previously found by Sahidullah and Saha (2012) to degrade performance.

One can see from Tables 4 and 5 that OE-MFCC and block MFCC appear to have better performance than MFCC. This becomes increasingly true when the number of fi in the fi banks are increased. OE-MFCC is better than block MFCC for most cases. This is especially true when a relatively large number of fi are in the fi bank(s). OE-MFCC is found to have superior performance and the lowest EER. For all OE-MFCC, block MFCC and MFCC, better features appear to be extracted with multitaper spectrum estimation.

<table>
<thead>
<tr>
<th>Filter Bank</th>
<th>MFCC Dim.</th>
<th>MFCC EER</th>
<th>OE Dim.</th>
<th>OE EER</th>
<th>NOBT Blocks</th>
<th>NOBT EER</th>
<th>NOBT Dim.</th>
<th>NOBT EER</th>
<th>OBT Blocks</th>
<th>OBT EER</th>
<th>OBT Dim.</th>
<th>OBT EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>57</td>
<td>4.25%</td>
<td>52</td>
<td>4.07%</td>
<td>1-8, 9-20</td>
<td>54</td>
<td>4.16%</td>
<td>1-9, 8-20</td>
<td>60</td>
<td>4.10%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>69</td>
<td>4.74%</td>
<td>64</td>
<td>4.30%</td>
<td>1-9, 10-24</td>
<td>66</td>
<td>3.88%</td>
<td>1-10, 9-24</td>
<td>72</td>
<td>4.19%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>81</td>
<td>4.97%</td>
<td>76</td>
<td>3.80%</td>
<td>1-11, 12-28</td>
<td>78</td>
<td>4.67%</td>
<td>1-12, 11-28</td>
<td>84</td>
<td>4.21%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Performance comparison of OE-MFCC, block MFCC and MFCC using Hamming window spectrum smoothing.

<table>
<thead>
<tr>
<th>Filter Bank</th>
<th>MFCC Dim.</th>
<th>MFCC EER</th>
<th>OE Dim.</th>
<th>OE EER</th>
<th>NOBT Blocks</th>
<th>NOBT EER</th>
<th>NOBT Dim.</th>
<th>NOBT EER</th>
<th>OBT Blocks</th>
<th>OBT EER</th>
<th>OBT Dim.</th>
<th>OBT EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>57</td>
<td>3.81%</td>
<td>52</td>
<td>4.06%</td>
<td>1-8, 9-20</td>
<td>54</td>
<td>4.32%</td>
<td>1-9, 8-20</td>
<td>60</td>
<td>3.79%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>69</td>
<td>3.90%</td>
<td>64</td>
<td>3.85%</td>
<td>1-9, 10-24</td>
<td>66</td>
<td>3.99%</td>
<td>1-10, 9-24</td>
<td>72</td>
<td>3.92%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>81</td>
<td>4.72%</td>
<td>76</td>
<td>3.16%</td>
<td>1-11, 12-28</td>
<td>78</td>
<td>3.84%</td>
<td>1-12, 11-28</td>
<td>84</td>
<td>3.99%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Performance comparison of OE-MFCC, block MFCC and MFCC using multitaper spectrum estimation.

5.4. Effect of Multitaper Type and Tapers Number in Multitaper-Fitted LPCC Features

Speaker recognition performance in the i-vector system is evaluated using the proposed multitaper-fitted LPCC features. The performance is investigated for the taper types previously used for MFCC extraction in (Kinnunen et al., 2010). These are the Thomson, Multipeak and sine tapers.
From Fig. 11, we see that the four multipeak tapers appear to achieve optimum performance with a minimum EER of 4.11%. The performance is compared to the baseline of the commonly used Hamming window. It can also be seen that a higher number of tapers (over 5) degrades performance due to an increase in side-lobes of higher order tapers. The results presented by multitaper-fitted LPCC and the results from (Kinnunen et al., 2010) and (Alam et al., 2013), confirm that multipeak tapers are the best multitaper type for speech processing in speaker recognition.

5.5. Feature Transformation

In this subsection, the effect of the transformation of MFCC and OE-MFCC features using the proposed RNN based PCA is investigated. According to the original feature dimensionality (see Table 3), the total number of principal components is 39 for MFCC and 76 for OE-MFCC. Four types of PCA were investigated and the results are listed in Table 6.

<table>
<thead>
<tr>
<th>d</th>
<th>CV-PCA</th>
<th>WCV-PCA</th>
<th>CR-PCA</th>
<th>WCR-PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>All (39)</td>
<td>3.89</td>
<td>4.08</td>
<td>3.82</td>
<td>4.17</td>
</tr>
<tr>
<td>35</td>
<td>3.74</td>
<td>2.96</td>
<td>3.51</td>
<td>3.28</td>
</tr>
<tr>
<td>34</td>
<td>3.96</td>
<td>3.00</td>
<td>2.96</td>
<td>2.81</td>
</tr>
<tr>
<td>33</td>
<td>3.67</td>
<td>2.71</td>
<td>2.97</td>
<td>2.79</td>
</tr>
<tr>
<td>32</td>
<td>3.64</td>
<td>3.28</td>
<td>2.91</td>
<td>2.97</td>
</tr>
<tr>
<td>31</td>
<td>3.56</td>
<td>3.23</td>
<td>3.34</td>
<td>2.81</td>
</tr>
<tr>
<td>30</td>
<td>3.47</td>
<td>3.47</td>
<td>3.21</td>
<td>3.37</td>
</tr>
<tr>
<td>29</td>
<td>3.96</td>
<td>3.50</td>
<td>3.63</td>
<td>3.38</td>
</tr>
<tr>
<td>28</td>
<td>3.79</td>
<td>3.46</td>
<td>3.31</td>
<td>3.09</td>
</tr>
<tr>
<td>27</td>
<td>3.61</td>
<td>3.10</td>
<td>3.64</td>
<td>3.41</td>
</tr>
<tr>
<td>26</td>
<td>4.20</td>
<td>3.69</td>
<td>4.12</td>
<td>3.56</td>
</tr>
<tr>
<td>AV</td>
<td>3.76</td>
<td>3.24</td>
<td>3.36</td>
<td>3.45</td>
</tr>
<tr>
<td>STD</td>
<td>0.22</td>
<td>0.29</td>
<td>0.37</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Table 6: System performance (in EER%) using transformed MFCC features.

6 and 7.
It can be observed that the performance using CV-PCA is comparable to the reference performance (Table 3). In fact, covariance PCA outperforms correlation PCA solely for dimensionality reduction. This is because it gives similar performance in relation to the one reported in the tables even for a lower number of components. It is also noticeable that WCV-PCA outperforms CV-PCA. However, WCR-PCA provides the lowest average EER in both cases.

Recall that odd and even subsets of a feature bank in OE-MFCC interchangeably capture the speech spectrum. That can cause their cepstral coefficients to be more correlated than conventional MFCC coefficients. These features could therefore benefit from an orthogonal transformation such as PCA.

Recall that odd and even subsets of a feature bank in OE-MFCC interchangeably capture the speech spectrum. That can cause their cepstral coefficients to be more correlated than conventional MFCC coefficients. These features could therefore benefit from an orthogonal transformation such as PCA.

<table>
<thead>
<tr>
<th></th>
<th>CV-PCA</th>
<th>WCV-PCA</th>
<th>CR-PCA</th>
<th>WCR-PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>All (76)</td>
<td>4.52</td>
<td>4.20</td>
<td>4.33</td>
<td>4.41</td>
</tr>
<tr>
<td>45</td>
<td>3.87</td>
<td>3.23</td>
<td>3.61</td>
<td>3.10</td>
</tr>
<tr>
<td>44</td>
<td>3.46</td>
<td>3.06</td>
<td>3.64</td>
<td>2.80</td>
</tr>
<tr>
<td>43</td>
<td>3.04</td>
<td>3.16</td>
<td>3.36</td>
<td>2.52</td>
</tr>
<tr>
<td>42</td>
<td>2.98</td>
<td>2.75</td>
<td>3.64</td>
<td>2.75</td>
</tr>
<tr>
<td>41</td>
<td>3.46</td>
<td>3.06</td>
<td>3.21</td>
<td>3.21</td>
</tr>
<tr>
<td>40</td>
<td>3.36</td>
<td>2.52</td>
<td>3.21</td>
<td>3.10</td>
</tr>
<tr>
<td>39</td>
<td>3.42</td>
<td>3.79</td>
<td>2.52</td>
<td>2.28</td>
</tr>
<tr>
<td>38</td>
<td>3.18</td>
<td>3.06</td>
<td>2.47</td>
<td>2.36</td>
</tr>
<tr>
<td>37</td>
<td>3.26</td>
<td>3.25</td>
<td>2.59</td>
<td>2.25</td>
</tr>
<tr>
<td>36</td>
<td>3.21</td>
<td>3.18</td>
<td>2.55</td>
<td>2.47</td>
</tr>
<tr>
<td>AV</td>
<td>3.32</td>
<td>3.11</td>
<td>3.08</td>
<td>2.68</td>
</tr>
<tr>
<td>STD</td>
<td>0.25</td>
<td>0.33</td>
<td>0.49</td>
<td>0.36</td>
</tr>
</tbody>
</table>

Table 7: System performance (in EER%) using transformed OE-MFCC features.

5.6. Feature Fusion

The performance of the system is evaluated for two combinations of features fused using PCA. These combinations are MFCC+LPCC and OE-MFCC+LPCC. The performance in terms of EER is reported in Tables 8 and 9.

Similar to the case of feature transformation, weighted PCA outperforms unweighted PCA; and PCA of the correlation matrix outperforms PCA of the covariance matrix. The average EER in the fusion of MFCC and LPCC is comparable to that of OE-MFCC which demonstrates the power of OE-MFCC. WCR-PCA appears to offer the best average performance for all feature combinations studied. This might be explained by the following reasons: 1) use of the correlation matrix instead of the covariance matrix; 2) using weights for the population feature vectors; and 3) using an iterative approach to determine the principal components. WCV-PCA appears to offer an enhancement over CV-PCA, while CV-PCA results in the highest average EER.

A notable aspect of the results presented here is that projection on all the principal components gives relatively high EER values. This is because having a relatively high feature dimensionality corresponds to a high number of principal components. The higher order principal components correspond to a low percentage of the variance. These principal
components may not only represent low variance but also noise or other distracting effects embedded in the original features. Projection on these principal components results in sets of attributes that are found to negatively affect the recognition performance.

The performance obtained from using all the principal components in all the investigated cases of transformation and fusion supports our explanation. Particularly in Table 9, which shows the EER for the case of fusing OE-MFCC and LPCC features. It is evident that using all the principal components (115) gives relatively high EER. This can be compared to the reference performance for the concatenation of OE-MFCC and LPCC features reported in Table 3. The reason is that, given the high feature dimensionality of 115 (76 dimensions of OE-MFCC plus 39 dimensions of LPCC), the higher order components, relatively, represent extremely low variance. Another noticeable aspect in the fusion of OE-MFCC and LPCC, is that even CV-PCA has a considerable reduction in EER. This means that a mere concatenation of different features is not as effective as anticipated. This is especially true when the accumulated feature dimensionality becomes relatively high.

Table 8: System performance (in EER%) using the fusion of MFCC and LPCC.

<table>
<thead>
<tr>
<th>d</th>
<th>CV-PCA</th>
<th>WCV-PCA</th>
<th>CR-PCA</th>
<th>WCR-PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>All (78)</td>
<td>4.04</td>
<td>3.92</td>
<td>4.05</td>
<td>4.19</td>
</tr>
<tr>
<td>45</td>
<td>4.05</td>
<td>3.10</td>
<td>3.52</td>
<td>2.90</td>
</tr>
<tr>
<td>44</td>
<td>3.55</td>
<td>2.79</td>
<td>3.31</td>
<td>3.00</td>
</tr>
<tr>
<td>43</td>
<td>3.55</td>
<td>3.07</td>
<td>2.77</td>
<td>2.71</td>
</tr>
<tr>
<td>42</td>
<td>3.31</td>
<td>2.79</td>
<td>2.47</td>
<td>2.41</td>
</tr>
<tr>
<td>41</td>
<td>3.31</td>
<td>3.05</td>
<td>3.18</td>
<td>2.33</td>
</tr>
<tr>
<td>40</td>
<td>3.21</td>
<td>3.33</td>
<td>2.51</td>
<td>2.61</td>
</tr>
<tr>
<td>39</td>
<td>3.26</td>
<td>2.97</td>
<td>3.31</td>
<td>2.19</td>
</tr>
<tr>
<td>38</td>
<td>2.56</td>
<td>2.40</td>
<td>3.32</td>
<td>2.52</td>
</tr>
<tr>
<td>37</td>
<td>3.31</td>
<td>2.79</td>
<td>3.18</td>
<td>2.71</td>
</tr>
<tr>
<td>36</td>
<td>3.85</td>
<td>2.95</td>
<td>3.31</td>
<td>2.35</td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AV</td>
<td>3.40</td>
<td>2.93</td>
<td>3.09</td>
<td>2.57</td>
</tr>
<tr>
<td>STD</td>
<td>0.40</td>
<td>0.25</td>
<td>0.36</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Table 9: System performance (in EER%) using the fusion of OE-MFCC and LPCC.

<table>
<thead>
<tr>
<th>d</th>
<th>CV-PCA</th>
<th>WCV-PCA</th>
<th>CR-PCA</th>
<th>WCR-PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>All (115)</td>
<td>6.03</td>
<td>5.72</td>
<td>5.86</td>
<td>5.90</td>
</tr>
<tr>
<td>50</td>
<td>3.10</td>
<td>2.68</td>
<td>2.71</td>
<td>2.66</td>
</tr>
<tr>
<td>49</td>
<td>3.26</td>
<td>2.32</td>
<td>2.71</td>
<td>2.11</td>
</tr>
<tr>
<td>48</td>
<td>2.99</td>
<td>2.59</td>
<td>2.54</td>
<td>1.91</td>
</tr>
<tr>
<td>47</td>
<td>3.12</td>
<td>2.36</td>
<td>2.36</td>
<td>2.35</td>
</tr>
<tr>
<td>46</td>
<td>2.91</td>
<td>2.22</td>
<td>2.71</td>
<td>1.99</td>
</tr>
<tr>
<td>45</td>
<td>2.91</td>
<td>2.19</td>
<td>1.99</td>
<td>2.08</td>
</tr>
<tr>
<td>44</td>
<td>3.23</td>
<td>2.50</td>
<td>2.11</td>
<td>1.97</td>
</tr>
<tr>
<td>43</td>
<td>2.86</td>
<td>2.22</td>
<td>2.71</td>
<td>1.89</td>
</tr>
<tr>
<td>42</td>
<td>3.01</td>
<td>2.01</td>
<td>2.53</td>
<td>1.97</td>
</tr>
<tr>
<td>41</td>
<td>2.96</td>
<td>2.97</td>
<td>2.40</td>
<td>2.17</td>
</tr>
<tr>
<td>AV</td>
<td>3.03</td>
<td>2.41</td>
<td>2.48</td>
<td>2.11</td>
</tr>
<tr>
<td>STD</td>
<td>0.13</td>
<td>0.28</td>
<td>0.26</td>
<td>0.23</td>
</tr>
</tbody>
</table>
Fig. 12 depicts the variability in EER in relation to using a variable number of principal components in the projection of the features. Compared to the performance of CR-PCA, the EER from WCR-PCA exhibits relatively low fl that result from varying the number of principal components. This similarly applies to WCV-PCA. The feature vector weighting process appears to reduce the significance of feature vectors that are more severely affected by noise or other non-speech sounds like breathing.

The best average EER achieved in this work is 2.11%, reduced from 3.76%. This was using development data that contained 639 male and 639 female speakers with 5112 utterances in total. Given the limited development data, the relative improvement in the performance is comparable to that of GMM-UBM/i-vector framework reported in (Khosravani and Homayounpour, 2018) with an EER of 1.13%. Note that in (Khosravani and Homayounpour, 2018), the development data contained 1925 male and 2603 female speakers which enabled a baseline performance of 2.40% EER.

PCA influence on the performance is judged based on the average of EER values. The average is calculated over a range of principal components used for feature transformation. It is difficult to anticipate if the performance for a particular number of principal components will be exactly the same in a different system. This is also true for a different evaluation set or with different data used to extract the principal components. However, the average EER exhibits a notable improvement over the performance with non-transformed features.

5.7. Computation Time

The complexity of the system described earlier has been reduced using the presented methods for PCA whilst the performance has been improved. The reduced feature dimensionality reduces the processing time taken by various elements of the system. The highest original feature dimensionality $D$ was 115 for the concatenation of OE-MFCC and LPCC features. In Table 9 we can see that this high feature dimensionality can be drastically reduced whilst retaining good performance. Dimensionality reduction down to $d = 43$ is shown in Table 10 as an example to show the reduction in computation complexity.
Table 10: Computation time for the processes affected by features dimension.

<table>
<thead>
<tr>
<th>Process</th>
<th>$D$</th>
<th>$d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM-UBM Estimation</td>
<td>3.4 ms</td>
<td>2.5 ms</td>
</tr>
<tr>
<td>Baum-Welch Statistics Calculation</td>
<td>150 ms</td>
<td>110 ms</td>
</tr>
<tr>
<td>Total Variability Learning</td>
<td>1480 ms</td>
<td>940 ms</td>
</tr>
<tr>
<td>i-vector Extraction</td>
<td>670 ms</td>
<td>250 ms</td>
</tr>
</tbody>
</table>

The processing time for the estimation of the GMM-UBM is reported as per feature vector. Similarly for the total variability subspace training, the processing time is reported per Baum-Welch statistics supervector. The reported time of calculating Baum-Welch statistics is for a 150 seconds long speech utterance. The most significant reduction in computation time is in the extraction of the i-vector. This suggests that a variety of speech features can be combined to improve speaker recognition performance using the proposed methodologies whilst retaining relatively low system complexity.

The processing time taken by the principal component analysis was also investigated. Fig. 13 illustrates the time taken by three PCA methods versus the dimensionality of the features and feature combinations. The processing time is presented per feature vector. Obviously, the iterative methods (for 50 iterations) take less time than the classical method (i.e. SVD). Also, the time taken by SVD increases for each additional dimension. The power iteration and the RNN methods have almost the same processing time. However, it can be seen in Fig. 7 that RNN requires fewer iterations to converge than the power iteration method. Hence, the RNN method is superior to power iteration in situations where the principal components require a higher number of iterations to converge (for example: 500 iterations) as in (Delchambre, 2014).
5.8. Evaluation on Youtube Data

This section presents an evaluation of the proposed methodologies on the VoxCeleb\textsuperscript{2} dataset recently introduced in (Nagrani et al., 2017). This dataset contains speech excerpts of 1251 celebrities collected from Youtube. The average utterance length is 8 seconds and the utterances of 40 speakers are reserved for evaluation. The evaluation here follows the official protocol which includes a total of 37720 verification trials. The sampling rate of the speech utterances is 16 KHz providing a broader spectrum (compared to telephone data) to exploit by the feature extraction algorithms. The frequency range 50-7000Hz is considered in the feature extraction in this work.

The VoxCeleb data offers a large amount of development utterances (148642 utterances from 1211 speakers). Only part of those utterances are used here to train the models of the i-vector/PLDA speaker verification framework. This is because it is time consuming to use all of the development utterances in estimating the total variability matrix (T) especially for the case of using full dimensional features. Specifically, we use 45 utterances from each of the 1211 development speakers. For speakers that have more than 45 utterances, the longest 45 utterances are used. For training the GMM-UBM model with 2048 mixtures, the feature vectors of the 54495 development utterances were down-sampled to $\sim 26$ hours worth of feature vectors. The same amount of feature vectors is used in the PCA. The i-vectors' dimensionality is 400 reduced to 150 using the LDA analysis. Therefore, the final dimensionality of the development and evaluation i-vectors is 150 which is also the size of the Gaussian PLDA subspace.

Table 11 shows the system performance using MFCC features, the proposed OE-MFCC and Multitaper-Fitted LPCC features. These results using the VoxCeleb data include the Minimum Detection Cost Function (minDCF) defined in (Nagrani et al., 2017) and the EER(%). One can notice that OE-MFCC features appear to provide the best performance in comparison to MFCC and LPCC features. Tables 12 & 13 show the results of dimensionality reduction and feature fusion using PCA and weighted PCA based on the correlation and the covariance matrices.

<table>
<thead>
<tr>
<th>Features (Filter Bank Size)</th>
<th>Feature Dimensionality</th>
<th>EER (%)</th>
<th>minDCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC (24)</td>
<td>39</td>
<td>8.55</td>
<td>0.70</td>
</tr>
<tr>
<td>OE-MFCC (28)</td>
<td>76</td>
<td>7.90</td>
<td>0.65</td>
</tr>
<tr>
<td>LPCC</td>
<td>39</td>
<td>10.29</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Table 11: Results on VoxCeleb dataset using MFCC features and the proposed OE-MFCC and Multitaper Fitted LPCC features. Enrollment and test i-vectors are scored using the Gaussian PLDA model.

Tables 12 & 13 show that the fusion of the proposed OE-MFCC and Multitaper-Fitted LPCC features improves the EER (%) and minDCF with all PCA forms considered (CR, WCR, CV and WCV). This is also considering the best performance in (Nagrani et al., 2017) using the DNN verification framework. Additionally, the best performance here, EER

\*\*By VoxCeleb we refer to the VoxCeleb\textsuperscript{1} dataset.
= 6.98% & minDCF = 0.57, is achieved with the fusion of the proposed OE-MFCC and Multitaper-Fitted LPCC features (see the case of WCV-PCA in Table 13). This result also outperforms the combination of OE-MFCC and LPCC features in a system’s score fusion fashion as shown in Table 14.

<table>
<thead>
<tr>
<th>Features</th>
<th>CR-PCA</th>
<th>WCR-PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EER (%)</td>
<td>minDCF</td>
</tr>
<tr>
<td>MFCC</td>
<td>9.27</td>
<td>0.73</td>
</tr>
<tr>
<td>OE-MFCC</td>
<td>7.77</td>
<td>0.68</td>
</tr>
<tr>
<td>MFCC+LPCC</td>
<td>8.39</td>
<td>0.67</td>
</tr>
<tr>
<td>OE-MFCC+LPCC</td>
<td>7.48</td>
<td>0.64</td>
</tr>
<tr>
<td>DeepCNN Framework</td>
<td>7.80</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Table 12: Results of the dimensionality reduction and feature fusion on experiments performed using the VoxCeleb dataset using correlation-matrix based PCA (CR-PCA) and weighted PCA (WCR-PCA). $d$ is the resultant feature dimensionality.

<table>
<thead>
<tr>
<th>Features</th>
<th>CV-PCA</th>
<th>WCV-PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EER (%)</td>
<td>minDCF</td>
</tr>
<tr>
<td>MFCC</td>
<td>8.70</td>
<td>0.67</td>
</tr>
<tr>
<td>OE-MFCC</td>
<td>7.16</td>
<td>0.64</td>
</tr>
<tr>
<td>MFCC+LPCC</td>
<td>7.76</td>
<td>0.62</td>
</tr>
<tr>
<td>OE-MFCC+LPCC</td>
<td>6.98</td>
<td>0.60</td>
</tr>
<tr>
<td>DeepCNN Framework</td>
<td>7.80</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Table 13: Results of the dimensionality reduction and feature fusion on experiments performed using the VoxCeleb dataset. Results for the covariance based PCA (CV-PCA) and weighted PCA (WCV-PCA) are shown. The resultant feature dimensionality is $d$.

<table>
<thead>
<tr>
<th>Features</th>
<th>Feature Dimensionality</th>
<th>EER (%)</th>
<th>minDCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC + LPCC</td>
<td>39 + 39</td>
<td>7.74</td>
<td>0.64</td>
</tr>
<tr>
<td>OE-MFCC + LPCC</td>
<td>76 + 39</td>
<td>7.32</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Table 14: Results on VoxCeleb dataset produced with the fusion of the system’s evaluation scores when MFCC, OE-MFCC and LPCC features are used. Equal score fusion weight is used.

It can be seen from Tables 12 & 13 that the proposed weighted PCA methodology outperforms conventional PCA for most cases. It can also be observed that covariance based PCA appears to provide better results than the correlation based PCA. This was occasionally the case in the evaluation on the Det5 telephone-speech subset. However, there is a number of factors that differ between the two evaluation sets (VoxCeleb and Det5).
It is difficult to precisely attribute this observation to a particular factor without further analyses. It might be rational to attribute it to the difference in the spectrum range exploited by feature extraction. Note that the sampling rate is 16 KHz for VoxCeleb and 8 KHz for Det5. The type and power of noise exhibited by each evaluation set might also affect the relative difference in performance.

However, the contributions made in this work are found to provide a positive impact on the performance using two different datasets. It is anticipated that further improvements can be seen on the VoxCeleb dataset with computational capacities that can make use of the entire development utterances.

6. Conclusions and Future Work

The work presented here shows that there is still room for research in the front-end of speaker recognition systems. To evaluate the proposed methodologies on telephone speech, data augmentation was necessary to establish the i-vector/PLDA speaker verification system. It is shown here that the performance of this system degrades noticeably when the development data is insufficient. If the system is already trained with sufficient development data then the augmentation may not present any noticeable improvement. However, it should be noted that it is unlikely to harm the performance. Also, the Gaussian noise may help in accounting for unknown mismatch between the enrolment and test samples; if it can be suitably modelled with a normal distribution.

When the input resembles a highly correlated Markov-I process then the DCT transform approximates the Karhounen-Loeve transform (Sahidullah and Saha, 2012). This is often considered important to reduce the losses in DCT transformation. The proposed OE-MFCC highlights the role of the filter bank(s) as a ‘transformation’ of the speech spectrum. This is in contrast to the DCT transformation of the filter outputs. It is shown here that using subsets of non-overlapped filters results in having a lower residual correlation. This is considered to be more important, in this work, since the filter bank design is based on the perceptual mechanism of speech. This hypothesis is confirmed by the experimental results which show that OE-MFCC outperforms MFCC.

The results reported here show that multitaper spectrum estimation appears to improve the performance obtained with MFCC, OE-MFCC and LPCC features. The proposed approximation of the autocorrelation function is based on smooth spectral estimates. The methodology could therefore have penalised rapid changes in the spectral envelop of high pitched speakers, as previously addressed by (Ekman et al., 2008). This can be properly investigated in future work.

The PCA based fusion of the features introduced here appears to provide the best performance on both of the datasets used in the evaluations. The proposed RNN framework for extracting the principal components is a relatively fast and an uncomplicated iterative process. It is found to be suitable for the eigendecomposition of weighted covariance and correlation matrices. The proposed weighting criterion appears to decrease the significance of bad and outlying feature vectors. SVD based initialisation of the principal components appears to help with the convergence rate of the RNN. The number of iterations specified,
50, should also allow the network to converge even if arbitrary vectors are used. Feature vector weighting remains an interesting area for future research where different weighting methodologies could be investigated.

Acknowledgements

The first author would like to thank the Higher Committee for Education Development in Iraq for funding his PhD study at the University of Portsmouth. The authors would like to thank the anonymous reviewers for their helpful comments.

References

Barnard, E., Davel, M., Van Heerden, C., 2009. ASR corpus design for resource-scarce languages. ISCA.


URL: https://catalog.ldc.upenn.edu/LDC2004S04


Dear Dr. Frédéric Bimbot,

Declarations of interest

The authors wish to declare no conflict of interest.

Yours faithfully,

Ahmed Isam Ahmed