Extracting Sub-glottal and Supra-glottal Features from MFCC using Convolutional Neural Networks for Speaker Identification in Degraded Audio Signals

Anurag Chowdhury and Arun Ross
Michigan State University
chowdh51@cse.msu.edu, rossarun@cse.msu.edu

Abstract

We present a deep learning based algorithm for speaker recognition from degraded audio signals. We use the commonly employed Mel-Frequency Cepstral Coefficients (MFCC) for representing the audio signals. A convolutional neural network (CNN) based on 1D filters, rather than 2D filters, is then designed. The filters in the CNN are designed to learn inter-dependency between cepstral coefficients extracted from audio frames of fixed temporal expanse. Our approach aims at extracting speaker dependent features, like Sub-glottal and Supra-glottal features, of the human speech production apparatus for identifying speakers from degraded audio signals. The performance of the proposed method is compared against existing baseline schemes on both synthetically and naturally corrupted speech data. Experiments convey the efficacy of the proposed architecture for speaker recognition.

1. Introduction

Speaker recognition entails determining or verifying the identity of a speaker from an audio sample [32]. Developing automatic speaker recognition systems that are robust to various types of audio degradations is an open research problem. Such systems can benefit a wide variety of applications ranging from e-commerce and personalized user interfaces to surveillance and digital forensics. Popular consumer products such as Alexa [2], Google Home [3], Siri [6], point to the acceptance and demand of speech and speaker recognition based technologies in the commercial market.

In this research, we focus on developing a method for text-independent speaker identification from degraded audio signals.

2. Related Work

Some of the early works [32] in text-independent speaker identification used Gaussian Mixture Models (GMM) for modeling individual speakers based on the mel-frequency cepstral coefficients (MFCC) of their speech data. The parameters of the GMM were estimated from the training speech data of each speaker using the Expectation Maximization algorithm.

The GMM based speaker recognition algorithm [32] was then extended using adapted Gaussian mixture models [31]. The main idea was to train GMM with a large number of mixtures (256 to 1024), called the Universal Background Model (UBM), to model speaker-independent features. Parameters of the UBM were then adapted using each enrolled speaker’s training speech data to generate speaker adapted GMMs.

In order to enable the use of simple metrics such as cosine similarity for speaker verification, the high-dimensional GMM supervectors were transformed to a lower-dimensional space, called the total variability space, using factor analysis [12]. A given speech utterance in this total variability space is represented by a low-dimensional vector called i-vector [13]. The i-vectors are then used for speaker verification.

Noise in speech signals is a common detrimental factor in the performance of speaker recognition algorithms. Several spectrum estimation methods [29, 16] and speech enhancement techniques [27] have been evaluated as front-end processing techniques for developing noise robust speaker recognition methods. Voice activity detection is a technique used for detecting parts of the audio with speech activity in them. Sadjadi et. al. [34] used it as a front-end processing technique for detecting and removing non-speech parts of the audio, which are typically long noisy audio segments.

Over the last decade, speaker recognition in noisy audio conditions has become an active area of research in the speaker recognition community and substantial progress has been made in the domain.
Additive background noise is the most common type of noise encountered in speech signals [28]. Spectral subtraction is a popular pre-processing technique to remove additive noise from the speech data. In [38], the authors propose a robust feature estimation method that can capture source and vocal tract related speech properties from spectral-subtracted noisy speech utterances. Authors use score level combination of MFCC and Wavelet Octave Coefficients of Residue (WOCOR) on the pre-processed audio for performing speaker recognition.

As in the case of speaker recognition from clean speech, MFCC features have been used for describing audio features of noisy speech signals. Most of the MFCC based methods use only the magnitude of the Fourier transform of the speech frames for performing speaker recognition. However, the authors in [37] have shown the benefits of using the phase information along with magnitude to improve the speaker recognition performance in noisy audio signals.

I-vectors have been extensively used for speaker recognition from noisy audio signals. Authors in [21] improve the robustness of i-vector based speaker verification system by introducing noisy training data when deriving the i-vectors and then applying probabilistic linear discriminant analysis (PLDA). In their experiments, the authors confirmed the efficacy of their technique by achieving significant gains in verification accuracy at various signal-to-noise ratio (SNR) levels. Furthermore, the above approach of using i-vector in conjunction with PLDA was verified to be the best performing algorithm even in “mismatched” noise conditions, where the noise characteristics in the training and testing set were different [24].

Another work [22], improved upon the effectiveness of i-vector based speaker recognition methods in the presence of noisy speech data by using first order Vector Taylor Series (VTS) approximation to extract noise-compensated i-vectors. The approach was inspired by the success of VTS in the field of automatic speech recognition to compensate for the nonlinear effects of noise in the cepstral domain.

Apart from additive noise, degradation also presents itself in form of convolutive reverberation in speech audio. The work in [42] addresses this issue in a two staged approach. The authors first use the noisy speech data to train a DNN classifier to produce a binary time-frequency (T-F) mask. The mask is then used to separate out the unreliable T-F units at each audio frame. The masked output audio is then evaluated using GMM-UBM speaker models, trained in reverberant environments, to perform speaker recognition.

Another problem associated with speech production in noisy environments is the Lombard effect, where the speakers involuntarily tend to increase their vocal effort to make themselves better audible in noisy environment. This leads to significant impact on the speaker dependent characteristics of their speech. Authors in [18] have further established the dependence of Lombard speech on noise type and noise level using a GMM based Lombard speech type classifier.

In recent years, deep learning techniques have been developed for a large number of classification tasks including speaker recognition [25]. Richardson et. al [33] discussed the training of deep neural networks on frames of spectral audio features (like MFCC) for performing speaker recognition on the input frame. Another approach [33], suggested using Deep Neural Networks (DNN) for extracting a feature set from the input audio frames and then using a secondary neural network classifier for performing speaker recognition using the DNN learnt features. Zhang et. al. [41] trained multi-layer perceptrons and Deep Belief networks for learning discriminative feature transformations, and de-reverberated features from noisy speech data affected with microphone reverberations.

In the following sections, we will discuss the 1-D convolution filter based convolutional neural network (CNN) we have designed for speaker recognition on considerably degraded speech data.

3. Rationale behind automatic speaker recognition

For performing automatic speaker recognition, it is important to first understand how human speech is generated at the source. For generating voiced speech sounds, the sound source is provided by periodic vibration of the vocal folds by a process known as phonation. For phonation to occur, the ratio of the air pressure below the glottis (sub-glottal) to air pressure above the glottis (supra-glottal) must exceed a certain positive value [1]. The shape and size of the vocal tract imparts individuality to a speaker’s voice characteristics. MFCC features, as discussed further in the section 4.3.1, have been extensively used for capturing acoustic features of human vocal tract, which we have incorporated in our approach to perform speaker recognition.

Our approach for solving the problem of speaker recognition uses a Convolutional Neural Network (CNN) uniquely designed to learn the speaker dependent characteristics from patches of MFCC audio features. The MFCC features are widely used in the speech and speaker recognition community as they represent the shape of the envelope of the power spectral density of the speech audio, which in turn is a manifestation of the shape of the human vocal tract.

4. Proposed Algorithm

In the proposed work, we use 1-D convolutional filters for learning speaker dependent features from MFCC features for performing speaker identification in degraded audio signals. We model the problem of speaker identification as an image classification problem and propose a CNN ar-
MFCC Frames


4.1. Speech Parametrization

MFCC features are very popular in the speech and speaker recognition community. A detailed account of the MFCC feature extraction process can be found in [30, 32]. We used the VOICEBOX [9] toolbox for extracting MFCC feature from the audio data. Our 40 dimensional MFCC feature vector comprises of 20 mel-cepstral coefficients that includes the zeroth order coefficient, and 20 first order delta co-efficients. The hamming window is used in the time domain and triangular filters are used in the mel-domain.

4.2. Data Organisation

The input audio clip is split into smaller clips, of fixed temporal expanse, called audio frames. The number of audio frames in the input audio clip is determined by the length of a frame and the frame stride. The length of an audio frame, \( n \), is a function of the sampling frequency, \( f_s \). In the VOICEBOX [9] toolbox, \( n \) is expressed as follows:

\[
  n = 2^\left\lfloor \log_2(0.035 \times f_s) \right\rfloor .
\]

The frame stride is chosen to be \( n/2 \). We extract 40-dimensional MFCC features per audio frame of an audio clip. Upon extracting the MFCC feature from an audio clip, we obtain a two dimensional feature matrix, which is referred to as MFCC feature strip in this work. Each MFCC feature strip is of size \( 40 \times F \), where \( F \) is the number of extracted frames. Since the length of the input audio could be of arbitrary length, we extract MFCC feature patches containing fixed number of audio frames from the MFCC feature strip of the audio clip. The patches are extracted using a moving window approach, where the size of the window is set to 200 frames and the stride value to 100 frames. A visual representation of the MFCC feature strip of a clean audio sample and its corresponding noisy versions can be seen in Figure 1. The MFCC feature patches in the training and test sets were modified by subtracting the corresponding average image from them, in order to zero-center the data. The modified MFCC feature patch of size \( 40 \times 200 \) is now used as a two dimensional data input to the CNN network architecture described below.

4.3. 1-D Convolution

A traditional CNN architecture consists of a sequence of layers. Each layer transforms the input data by applying layer specific operations on the input and passing it over to the next layer. The three most common layer types found in a CNN architecture are: Convolutional Layer, Pooling Layer and Fully-Connected Layer. The convolutional layer in a CNN is where majority of the learning process takes place. Design and placement of the filters along the various layers of a CNN determine the “concepts” that are learned at each layer.

Deciding the shape of filters in CNNs is crucial to effectively learning the target concept from the input data. As discussed in [23], small square shaped filters are especially good for learning local patterns in image data, such as edges and corners, due to the high correlation between pixels in a small local neighborhood. However, that is not the case in the context of MFCC feature strips, as there is no local semantic structure (to our knowledge) that can be captured by a 2-D filter. As represented in Figure 1, the pixel values along Y axis corresponding to the MFCC features are on a logarithmic scale, while the pixel values along X axis corresponding to the time domain are on a linear scale. Hence a 1-D filter is better at learning speaker dependent characteristics from the MFCC features placed along the Y axis.

4.3.1 Sub-glottal and Supra-glottal features

In the field of speech recognition, 1-dimensional filters across the time variable have shown promising results [40] by effectively learning temporal characteristics in the data. However, in the context of text independent speaker recognition, the temporal relevance of speaker dependent characteristics across MFCC feature frames is greatly reduced (but not eliminated), as the content of the speech has often no bearing on the identity of the speaker (especially in cases where the data is collected in a controlled lab environ-
Figure 2. An illustration of the proposed speaker identification algorithm using 1-D CNN. The input MFCC feature strip is split into MFCC patches and evaluated on the trained CNN. The classification scores from different patches are fused to arrive at a classification decision.

4.4. ReLU NonLinearity and Pooling layers

The filter responses from each of the convolutional layers are made to pass through ReLU non-linearity as, unlike sigmoid activation functions, they do not suffer from the problem of vanishing gradients. Further, we used max-pooling to reduce the size of the parameter space to be learnt by the network.

4.5. Dropout

Dropout layers were added to introduce regularization in the CNN being trained. It provides the dual benefit of making the CNN robust towards perturbations in the input data while also mitigating the problem of over-fitting to the training data.

4.6. Score level fusion and Decision

In the testing phase, as illustrated in the Figure 2, the input MFCC feature strip, \( X \), is split into MFCC patches, \( x_i, i \in \{1, 2, 3, ..., N\} \), where, \( N \), is the number of patches. For every input MFCC patch, \( x_i \), the CNN gives a set of classification scores, \( \{s_{i,j}\}, j \in \{1, 2, 3, ..., C\} \), corresponding to the \( C \) speakers (e.g., \( C = 168 \) in the TIMIT and NTIMIT test datasets). Here, \( s_{i,j} \), is the classification score assigned to the \( j^{th} \) speaker for the \( i^{th} \) patch.

Scores from all the patches extracted from the audio clip are then added to give fused classification scores, \( \{S_j\} \), for the entire audio clip:

\[
S_j = \sum_{i=1}^{N} s_{i,j}, \forall j.
\]

The input audio is then assigned to the speaker \( j^* \) where,

\[
j^* = \arg \max_j \{S_j\}.
\]

Figure 3. Architecture of the CNN used for Speaker Identification from degraded audio samples. The input is a $40 \times 200 \times 1$ MFCC feature patch to the CNN. The last layer gives a classification score to each of the 168 speakers in the testing set in the TIMIT and NTIMIT datasets.

Table 1. Identification Results on the SITW, NTIMIT and Noisy variants of TIMIT speech dataset.

<table>
<thead>
<tr>
<th>Exp. #</th>
<th>Training set</th>
<th>Testing Set</th>
<th>Accuracy (Rank 1 in %)</th>
<th>Accuracy (Rank 5 in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>UBMM GMM</td>
<td>i-vector PLDA</td>
</tr>
<tr>
<td>1</td>
<td>Babble, F16, R1, V1</td>
<td>Car, Factory, R2, V2</td>
<td>1.98</td>
<td>32.93</td>
</tr>
<tr>
<td>2</td>
<td>Car, Factory, R2, V2</td>
<td>Babble, F16, R1, V1</td>
<td>10.61</td>
<td>35.61</td>
</tr>
<tr>
<td>3</td>
<td>Babble, Car, R2, V2</td>
<td>Babble, F16, Factory, R1, V1</td>
<td>14.08</td>
<td>47.61</td>
</tr>
<tr>
<td>4</td>
<td>F16, Factory, R1, V1</td>
<td>Babble, Car, R2, V2</td>
<td>4.86</td>
<td>38.59</td>
</tr>
<tr>
<td>5</td>
<td>Car, F16, R1, V1</td>
<td>Babble, Factory, R2, V2</td>
<td>3.27</td>
<td>21.13</td>
</tr>
<tr>
<td>6</td>
<td>Babble, Factory, R2, V2</td>
<td>Car, F16, R1, V1</td>
<td>3.86</td>
<td>24.60</td>
</tr>
<tr>
<td>7</td>
<td>NTIMIT</td>
<td>NTIMIT</td>
<td>70</td>
<td>71.11</td>
</tr>
<tr>
<td>8</td>
<td>SITW</td>
<td>SITW</td>
<td>83.33</td>
<td>14.08</td>
</tr>
</tbody>
</table>

5. Experiments

5.1 Datasets


5.1.1 TIMIT Dataset

The TIMIT dataset provides clean speech recordings of 630 speakers. There are 462 speakers in the training set and 168 speakers in the testing set. The dataset contains of eight major dialects of American English. There are ten sessions of 3 seconds each (so 10 audio samples) per speaker in the dataset. The text spoken by the speakers in the training set and test set are disjoint, making the speaker recognition experiments text-independent.

In our experiments, TIMIT dataset was perturbed [5, 19] with synthetic noise of different types (given below) from the NOISEX-92 [36] noise dataset. The noisy versions of the TIMIT dataset were generated in simulated room environments with different acoustic properties and reverberation levels, thereby introducing convoluted reverberations into the noise profile. The synthetically generated noisy datasets have the following noise characteristics:

1. Noise Type: Following four types of noises were added to the TIMIT dataset:
   1.1. F-16: Noise generated by engine of F-16 fighter aircraft.
   1.2. Babble: Noise generated by rapid and continuous background human speech.
   1.3. Car: Noise generated by engine of a car.
   1.4. Factory: Noise generated by heavy machinery operating in a factory environment.

2. Signal to Noise Ratio (SNR): The resultant noisy datasets were each generated at three different SNR levels, viz., 20 dB, 10dB and 0dB.

3. Room Size: The noisy dataset were generated in a simulated room environment with two different room sizes (4m and 20m, side length of cube), referred to as R1 and R2 in the protocol.

4. Reverberation: Two different reverberation coefficients were used to introduce additional noise in the data, referred to as V1 and V2 in the protocol.

5.1.2 Fisher English Training Speech Part 1 dataset

The Fisher English Training Speech Part 1 Speech dataset contains conversational speech data collected over telephone channels between pairs of speakers. This dataset has over 12, 000 speakers. Conversations pertaining to a subset of 1, 052 speakers from the Fisher dataset were chosen for the experiments in this work. Audio pertaining to each speaker in the conversation is then segmented out and processed with voice activity detection to remove empty audio segments from the audio. The audio of each speaker was then split into smaller audio snippets of around 3-second duration each. We extract 60 audio snippets for each speaker from their conversational audio.
Table 2. Identification Results on the Noisy variants of TIMIT speech dataset in presence of the extended gallery-set (1052+168 speakers). The extended gallery consists of audio samples from the Fisher speech dataset also.

<table>
<thead>
<tr>
<th>Exp. #</th>
<th>Training set</th>
<th>Testing Set</th>
<th>Accuracy (Rank 1 in %)</th>
<th>Accuracy (Rank 5 in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>UBMs-GMM</td>
<td>i-vector-PLDA</td>
</tr>
<tr>
<td>1</td>
<td>Babble, F16, R1, V1</td>
<td>Car, Factory, R2, V2</td>
<td>1.58</td>
<td>1.09</td>
</tr>
<tr>
<td>2</td>
<td>Car, Factory, R2, V2</td>
<td>Babble, F16, R1, V1</td>
<td>1.09</td>
<td>2.87</td>
</tr>
<tr>
<td>3</td>
<td>Babble, Car, R2, V2</td>
<td>F16, Factory, R1, V1</td>
<td>1.78</td>
<td>5.15</td>
</tr>
<tr>
<td>4</td>
<td>F16, Factory, R1, V1</td>
<td>Babble, Car, R2, V2</td>
<td>1.88</td>
<td>0.99</td>
</tr>
<tr>
<td>5</td>
<td>Car, F16, R1, V1</td>
<td>Babble, Factory, R2, V2</td>
<td>0.50</td>
<td>0.19</td>
</tr>
<tr>
<td>6</td>
<td>Babble, Factory, R2, V2</td>
<td>Car, F16, R1, V1</td>
<td>16.56</td>
<td>6.54</td>
</tr>
</tbody>
</table>

5.1.3 NTIMIT Dataset

NTIMIT [20] dataset consists of speech from the TIMIT dataset that was transmitted and re-collected over a telephone network. The speech content and speakers in the NTIMIT dataset are identical to that of the TIMIT dataset. But since the NTIMIT is collected over a telephone network, it has noise characteristics inherent to the telephone channel, thereby resulting in a noisy version of the TIMIT dataset. Even though the average SNR of NTIMIT dataset is higher (36dB) than that of the noisy versions of the TIMIT dataset that we had created (section 5.1.1), the former provides a much more realistic noise profile.

5.1.4 Speakers in the Wild (SITW) Database

The Speakers in the Wild (SITW) dataset [26] contains speech samples collected from open-source media for benchmarking and evaluating text-independent speaker recognition algorithms. Since the SITW data was not collected in a controlled setting, it contains real noise, reverberation, intra-speaker variability and compression artifacts. There are 299 speakers in the dataset (119 in the training set and 180 in the testing set) with variable number of audio samples of differing lengths per speaker. Audio of each speaker from the dataset is processed with voice activity detection to remove any empty audio segments. The audio for each speaker was then split into smaller audio snippets of around 3-second duration each. We extract 10 audio snippets for each speaker from their conversational audio.

5.2. Experimental Protocols

In the experiments involving noisy variants of the TIMIT dataset, we ensure disjoint noise characteristics in the training and testing sets as shown in Table 1. For example, in experiment 1, the training set consists of audio samples that are simulated to be recorded in a room of size R1 and reverberation coefficient V1, with additive background noise of type “Babble” and “F16”.

Apart from the six experiments on the noisy TIMIT datasets, we also perform speaker identification experiments on the NTIMIT and SITW datasets. The training and the testing sets in the NTIMIT dataset share the same noise profile (that of telephone channels), unlike the disjoint noise profiles in the noisy versions of TIMIT dataset created by us. The noise content in the SITW dataset varies greatly over samples both within and between different speakers.

Additionally, we also extended the six experiments on the noisy TIMIT datasets by adopting an extended gallery set comprising of a subset of 1052 speakers from the Fisher dataset alongside the original 168 speakers in the testing set of the TIMIT dataset. The extended gallery set, therefore, has 1220 speakers.

5.2.1 UBMs-GMM [31] based Speaker Identification

To obtain baseline performance on the eight experiments laid out in Table 1, we train a Universal Background Model (UBM) [31] using data from the speakers in the training set. The trained UBMs are then adapted using data from the speakers in the test set, to obtained speaker-adapted GMM models. For adapting the UBMs to individual speakers, nine audio samples per speaker is used, and the remaining audio sample per speaker is reserved for testing.

5.2.2 i-vector-PLDA [15] based Speaker Identification

To obtain a second baseline performance on the eight experiments laid out in Table 1, we train an i-vector-PLDA based speaker recognition system as implemented in the MSR identity toolkit [35]. Similar to the protocol for the UBM-GMM experiment, we use nine audio samples per speaker from the testing set for adapting the i-vector models, and the remaining audio sample per speaker is reserved for evaluation.

5.2.3 1-D CNN based Speaker Identification

The eight experiments, given in Table 1, were then conducted using the proposed 1-D CNN based Speaker Identification algorithm. Since the CNN based algorithm does not require a background model unlike UBM-GMM [31], we directly train the CNN on the speakers in the test set, with nine audio samples per speaker. The remaining audio sample per speaker is used in the test set.
5.2.4 Extended Gallery Speaker Identification

The six experiments, given in Table 2, are the extended gallery experiments that were done to test the discriminative power of the algorithms in presence of an extended gallery set. The speaker recognition models in the six extended-gallery experiments were trained in exactly the same way as they were done for the first six experiments in Table 1. The gallery set of 168 speakers from the TIMIT dataset are augmented with a subset of 1052 speakers from the Fisher English Training Speech Part 1 Speech dataset. The probe data is sourced from only the 168 speakers in the TIMIT dataset. Therefore, for each probe sample, the algorithms now have to make a decision from a pool of 1220 speakers, where 168 are from the TIMIT dataset and 1052 are from the Fisher dataset.

6. Results and Analysis

The results of the identification experiments are given in Tables 1 and 2. Both Rank-1 and Rank-5 identification accuracies (in %) are reported for the baseline methods and the proposed method. The Cumulative Match Characteristic (CMC) curves are given in Figures 4 and 6.

- The identification accuracy of the 1-D CNN based speaker identification algorithm is vastly superior at Rank 1 across all eight experiments in Table 1.
- The average identification accuracy across the first six experiments on the noisy TIMIT datasets is 33.40% at Rank 1 and 62.40% at Rank 5 for 1-D CNN, 9.29% at Rank 1 and 20.92% at Rank 5 for UBM-GMM and 7.56% at Rank 1 and 20.94% at Rank 5 for i-vector-PLDA.
- In the experiments on NTIMIT dataset, it is important to note that i-vector-PLDA outperforms UBM-GMM at both Rank 1 and Rank 5 indices, and it also outperforms the proposed 1-D CNN based algorithm at Rank 5. This could be attributed to the fact that i-vector-PLDA outperforms UBM-GMM in low noise scenarios and, since the NTIMIT dataset has higher average SNR (36dB) compared to that of the noisy variants of TIMIT dataset (10dB), the i-vector-plda performs better on the NTIMIT dataset. Even though the i-vector-PLDA outperforms 1-D CNN at Rank 5, it should be noted that 1-D CNN significantly outperforms i-vector-PLDA at Rank 1.
- In the STITW dataset, 1-D CNN based algorithm modestly outperforms the baseline algorithms at Rank 1.
- In the extended gallery experiments, the accuracy of 1-D CNN based speaker identification algorithm continues to be superior at both Rank 1 and Rank 5 indices across all six experiments. It is noteworthy that in experiment 5, UBM-GMM has a 0% accuracy at both Rank 1 and Rank 5, as it completely failed to identify the correct speakers at lower ranks in the extended gallery set. This substantiates the challenges of performing speaker identification in large datasets.
- On average, across the first six experiments in Table 1, UBM-GMM, i-vector-PLDA and 1-D CNN correctly identify the same 0.14% of the test samples at Rank 1. 1-D CNN correctly identifies an additional 26.60% of the test samples over both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1. However, the 1-D CNN based algorithm fails to correctly identify 2.64% of the test samples that were correctly identified by both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1.
- In the seventh experiment in Table 1, on the NTIMIT dataset, UBM-GMM, i-vector-PLDA and 1-D CNN based algorithms correctly identify the same 41% of the test samples at Rank 1. The 1-D CNN based algorithm correctly identifies an additional 10% of the test samples over both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1. However, 1-D CNN based algorithm fails to correctly identify 11% of the test samples that were correctly identified by both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1.
- In the eighth experiment in Table 1, on the SITW dataset, UBM-GMM, i-vector-PLDA and 1-D CNN based algorithms correctly identify the same 41.11% of the test samples at Rank 1. The 1-D CNN based algorithm correctly identifies an additional 0.06% of the test samples over both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1. However, the 1-D CNN based algorithm fails to correctly identify 0.02% of the test samples that were correctly identified by both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1.
- For the experiments with the extended gallery set in Table 2, on average, all three algorithms, UBM-GMM, i-vector-PLDA and 1-D CNN, correctly identified the same 0.82% of the test samples at Rank 1. 1-D CNN correctly identifies an additional 31.46% of the test samples over both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1. However, the 1-D CNN based algorithm fails to correctly identify 0.16% of the test samples that were correctly identified by both the UBM-GMM and i-vector-PLDA based algorithms at Rank 1. This establishes the superior discriminative power of the 1-D CNN based algorithm over both the baseline algorithms.
- In both the baseline algorithms and proposed algorithm, the MFCC features are used as input; but the performance of the 1-D CNN vastly improves over that of the baselines. This suggests that the 1-D CNN is better at extracting important speaker dependent characteristics, like sub-glottal and supra-glottal features, in presence of audio degradations.

7. Conclusion and Future Goals

Degradations in speech audio can distort and mask the speaker dependent characteristics in the audio signal. Tra-

8. Acknowledgement

This applied research effort was supported by the Federal Bureau of Investigation Biometric Center of Excellence.

References


