Int J Speech Technol (2017) 20:1005–1016
DOI 10.1007/s10772-017-9466-5

# Dravidian language classification from speech signal using spectral and prosodic features

Shashidhar G. Koolagudi[1] · Akash Bharadwaj[1] · Y. V. Srinivasa Murthy[1] · Nishaanth Reddy[1] · Priya Rao[1]

Received: 29 May 2017 / Accepted: 19 September 2017 / Published online: 14 October 2017

**Abstract** The interesting aspect of the Dravidian languages is a commonality through a shared script, similar vocabulary, and their common root language. In this work, an attempt has been made to classify the four complex Dravidian languages using cepstral coefficients and prosodic features. The speech of Dravidian languages has been recorded in various environments and considered as a database. It is demonstrated that while cepstral coefficients can indeed identify the language correctly with a fair degree of accuracy, prosodic features are added to the cepstral coefficients to improve language identification performance. Legendre polynomial fitting and the principle component analysis (PCA) are applied on feature vectors to reduce dimensionality which further resolves the issue of time complexity. In the experiments conducted, it is found that using both cepstral coefficients and prosodic features, a language identification rate of around 87% is obtained, which is about 18% above the baseline system using Mel-frequency cepstral coefficients (MFCCs). It is observed from the results that the temporal variations and prosody are the important factors needed to be considered for the tasks of language identification.

**Keywords** Dravidian language classification · Mel-frequency cepstral coefficients · Shifted delta cepstral features · Artificial neural networks · Legendre polynomial · Principle component analysis · Prosody analysis · Indian languages · Language identification

## 1 Introduction

Language identification is a trendy research problem for many years. In fact, there is a biannual competition named the National Institute of Standards and Technology (NIST) Language Recognition (LRE) competition which is running since 1996, resulting in a good number of publications dealing with the problem of language identification from speech (Torres-Carrasquillo et al. 2010; Matejka et al. 2006; Singer et al. 2012). There has also been a fair amount of research done using other data sets such as VoxForge data set (Montavon 2009; Marc 1996). However, the major thrust of most recent research is on improving the performance of a NIST-LRE task.

Mainstream research in language identification has involved very few Indian languages, Hindi and Tamil notably. India is a multi lingual country which has 22 official languages (based on the majority of speakers' speaks) and 1650 unofficial languages. All the 22 official languages contain more than a million of speakers (Chandrasekaran 2012; Jain and Cardona 2007). Among the Indian languages, Dravidian languages such as *Kannada, Malayalam, Tamil* and *Telugu* comes under the category spoken by the south Indian people. It is also true that the process of identifying them is complex since they share many commonalities. For

✉ Y. V. Srinivasa Murthy
urvishnu@gmail.com

Shashidhar G. Koolagudi
koolagudi@yahoo.com

Akash Bharadwaj
akash.bharadwaj92@yahoo.com

Nishaanth Reddy
nish.h.reddy@gmail.com

Priya Rao
priya.rao.62092@gmail.com

[1] Department of CSE, National Institute of Technology Karnataka, Mangalore 575 025, India

instance, Kannada and Telugu share the same script, and thus they have same phoneme set. Moreover, Kannada and Telugu have adopted a fair number of *Sanskrit* words as their own and they share 80% of the phonemes from *Devanagari*. On the other hand, Tamil evolved independently from Sanskrit. Malayalam is the language spoken by Kerala people and believed to have evolved from Tamil. All these issues and similarities lead to significant challenges for the correct identification of Dravidian languages. Tackling these challenges is the motivation for the work presented in this paper. The most successful approach in language identification is parallel phoneme recognition and language modeling (PPRLM) which generates a stream of phonemes from the language. It uses *n-gram* language models to model the phonotactics. It is computationally expensive and not suitable for the languages where they share same phonemes such as *Kannada* and *Telugu*. Hence, feature based approach has been proposed with cepstral coefficients such as Mel frequency cepstral coefficients (MFCCs), shifted delta cepstral (SDC) values and their combination with prosodic features and the same are considered for this work. To map the non-linearity being observed among the selected languages ANN is used.

The rest of the paper is laid out as follows: In Sect. 2 relevant literature is reviewed and a theoretical framework for our contribution is established. Section 3 elaborates the proposed methodology which includes the details of the database, feature extraction process, and methods used in this work. The results of these experiments are presented in Sect. 4. Section 5 concludes with a look towards future applications and extensions of this work.

## 2 Literature survey

When it comes to the task of language identification, there are three approaches that have been explored in past literature: The first one is the phonotactic method of language identification, known as phoneme recognition and language modeling (PRLM). This method involves identification of phonemes in speech and putting together groups of 2 or 3 consecutive phonemes (bi-grams or trigrams respectively). Each such group is then looked up in a dictionary for recognizing its language. This method is analogous to how humans perceive and understand spoken language. Humans put together such phonemes and identify them as words, which more often than not are unique to a language. In literature, this method has been shown to yield better results (Matejka et al. 2005; Zissman 1995; Li and Ma 2005). However, this performance comes at a price. Using a phonotactic system requires the use of computationally intensive procedures for identification of phonemes (Pinto et al. 2008; Graves et al. 2013). Moreover, such procedures are usually specific to each language. So, it is important to know which language the speech clip is in (which is not related to language identification task). For the above reasons, there has been a trend in literature to replicate the performance of phonotactic systems using other methods, for instance, GMM tokenizer (Torres-Carrasquillo et al. 2002). This method speeds up the process, however, it fails to meet the PRLM performance in identifying the language.

The second approach involves the use of spectral features such as MFCCs and their associated features such as SDCs (Shifted Delta Cepstral coefficients) (Torres-Carrasquillo et al. 2002; Allen et al. 2005). When it comes to speech, sounds generated by humans are modulated by the shape of vocal tract, the tongue and so on. This shape determines the type of sound or phoneme produced. The shape of oral cavity manifests itself in the envelope of short-time power spectrum of the speech signal. The extraction procedure of MFCCs captures exactly this. Further, SDC based features are used with GMM-UBM model for the task of language identification (Torres-Carrasquillo et al. 2002, 2010). Recent experiments done by extracting deep bottleneck features for identifying the spoken language shows that neural networks are also useful for the task of language identification (Jiang et al. 2014).

The third approach for language identification uses prosodic information such as syllable duration, rhythm, pitch and energy parameters etc. The idea of using prosody is essential and there are a few recent works that have explored the viability of prosody for the task of language identification (Raymond et al. 2010; Martínez et al. 2012). In this work, investigation is continued on this line of thought. It is found that for identification of Dravidian languages, prosody is a significant factor. We make a distinction between natural speech and read speech, where we find that prosody is a significant feature for identification of natural speech. The reason for this is, Dravidian languages majorly share a common vocabulary and script. However the major distinction is in speaking pattern that includes intonation, prolongation, nasalization, stress, pattern and so on. In addition to prosody zero crossing rate (ZCR) also found to be helpful in recognizing the language.

In addition to the approaches mentioned above, there are a few works designed based on i-vector, deep neural networks (DNNs), convolutional neural networks (CNNs), and so on. The features such as i-vector front-end features are found to be well suited for speaker verification systems. The works of language identification (LID) have motivated to use them as LID some times depends on the speaker as well (Brümmer et al. 2012; Li et al. 2013; Sturim et al. 2011). The whole utterance is considered as an input to form an i-vector which is further, optimized to a feature vector of size 400–600 dimensions (Dehak et al. 2011a, b). However, i-vector representation has a major drawback that they may not be suitable if the input is shorter utterance (Lopez-Moreno et al. 2016). The present research is highly motivated to utilize the advantages of deep neural networks (DNNs) as they found to outperform over the present era. They have been well used in visual object recognition (Ciresan et al. 2010), acoustic modeling (Hinton et al. 2012; Mohamed et al. 2012), natural language processing (Collobert and Weston 2008), and many other fields (Deng and Dong 2014). The basic system has been found with DNNs by Lopez-Moreno et al. (2014). They have considered passing the entire speech signal of various languages through DNN and found better accuracy when compared to the i-vectors. One more advantage is that they are capable of handling a large amount of data, unlike other approaches. The same causes complexity issues and not suitable for small data. The combination of DNN with i-vectors has also been experimented to achieve better performance (Ranjan et al. 2016). Later, they modified the number of words per phase to train the DNN. The use of convolutional neural networks (CNNs) is also found in some works to identify the language (Ganapathy et al. 2014). The majority of the works mentioned above are mostly based on the phoneme or word structure. Moreover, they need a massive amount of data for experimentation. Since the phonemes are almost similar in the case of Dravidian languages, it is necessary to think outside the box.

Dravidian languages are a part of south Indian languages that contains four different languages and can be classified them as two sets named as {Kannada, Telugu}, and {Malayalam, Tamil}. They share many commonalities such as script, phoneme structure, pronunciation, grammar, etc. Moreover, all the above four languages are evolved from Sanskrit. The PPRLM is a model which is designed mainly based on phonemes of different languages. Since Dravidian languages are sharing same phoneme structure, it is difficult to classify them using PPRLM.

In this experiment, prior to the feature extraction, a novel silence removal algorithm is proposed to avoid unuseful silence portion. Later, the combination of spectral and prosodic features is used with the ANN classifier to improve the performance compared to the existing work. Instead of using all prosodic features, polynomial fitting is done to select the representative features. Principle component analysis (PCA) is used for reduction of dimensionality which leads to less computational time. Result analysis is done on the real time speech taken from telephonic recordings instead of speech recorded in studio environments.

## 3 Proposed methodology

The process of the proposed work for Dravidian language classification system (DLCS) is shown in Fig. 1. The process starts with database collection, followed by preprocessing, feature extraction, optimization, and comparison of results using original and optimized feature vectors. The method proposed in this paper makes use of two different sets of features used for the purpose of Dravidian language Identification, namely, spectral and prosodic features. As spectral features, the first 13 Mel Frequency Cepstral Coefficients (MFCCs) are extracted from the speech frame. MFCC features are the quintessential features for any human speech processing task and are typically used to implement baseline systems for a wide array of problems, including language identification. Since SDCs have shown their importance in language identification and are useful for estimating the temporal variation, seven SDC coefficients are added to these MFCC features to evaluate the variation in the performance.

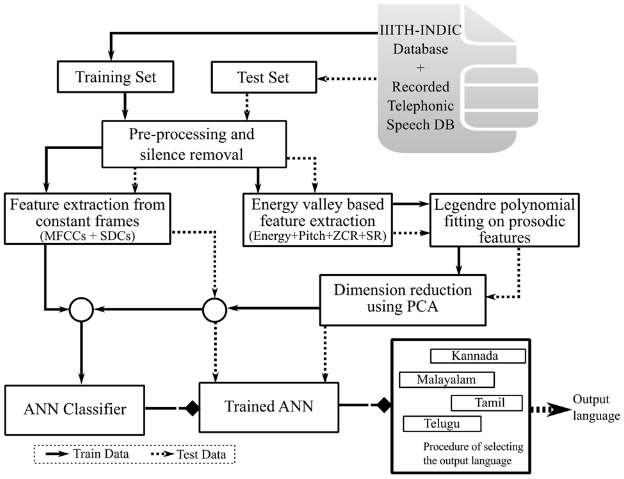

**Fig. 1** The proposed methodology for Dravidian language classfication system

It is a well-known fact that different languages have different pronunciation patterns, spoken at different speeds, involve different levels of stress or emphasis on phonemes during speaking lead to pitch variations and so on (Atal and Rabiner 1976). Indeed, these factors capture the dynamics of human speech that add a unique quality to each persons and (as it is proposed in this work) each languages speech representation. The specifics of feature extraction shall now be presented, followed by details about the actual classification phase.

### 3.1 Corpora

Two different data sets have been considered for the experiments conducted in this work. The first one is IIIT-Hyderabad INDIC data set (Prahallad et al. 2012). This database is mainly designed to motivate the researchers to work on language processing. At present, it has the speech clips for selected languages of Indian subcontinent such as Bengali, Hindi, Kannada, Malayalam, Marathi, Tamil and

**Table 1** The complete details of IIT-H INDIC database

| Sl. no. | Language | Region | No. of utterances | No. of speakers | Duration (min) |
|---|---|---|---|---|---|
| 1. | Kannada | Karnataka | 1000 | 5 | 95 |
| 2. | Malayalam | Kerala | 1000 | 5 | 103 |
| 3. | Tamil | Tamilnadu | 1000 | 5 | 92 |
| 4. | Telugu | Andhra Pradesh | 1000 | 5 | 110 |

**Table 2** The complete details about the collected dataset

| Sl. no. | Language | Region | No. of utterances | Duration (in min) |
|---|---|---|---|---|
| 1. | Kannada | Karnataka | 1500 | 250 |
| 2. | Malayalam | Kerala | 1500 | 250 |
| 3. | Tamil | Tamilnadu | 1500 | 250 |
| 4. | Telugu | Andhra Pradesh | 1500 | 250 |

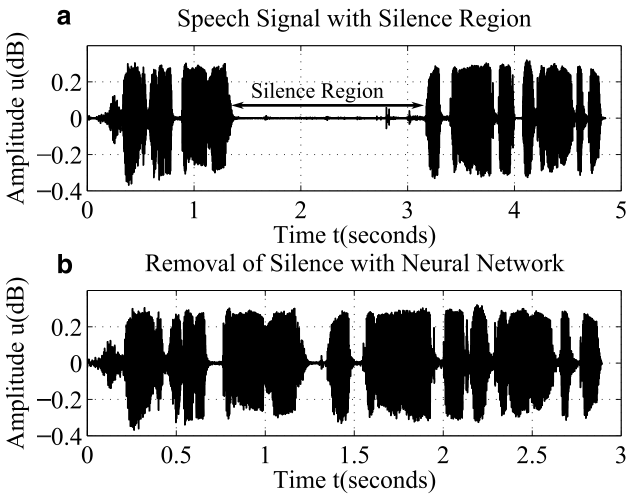

**Fig. 2** An example signal for representing the silence removal: **a** original speech signal and **b** speech signal after silence removal

Telugu. Since the experiments are related to Dravidian languages, the relevant data set has been separated. Each language of four contains 1000 sentences spoken by selected five native speakers based on their pleasantness in voice. The average length of the clip is around 6 s. Common words are identified based on the general requirements of humans. The speech has been recorded in the studio environment, and care has been taken to avoid noise and other distractions. It is important to note the speech looks like a read speech in this data set and do not contain many prosodic variations. Further details of the IIIT-Hyderabad dataset has been given in Table 1.

The second data set is manually collected from the TV and radio programs. An effort has been put to collect the dataset based on the audio recordings of radio and television chat programs. The recordings are for each of the four Dravidian languages mentioned above. The television shows deal with diverse topics to remove any dependencies on topic specific dialects. All the recordings are a speaker, gender, and content independent. The natural speech (as opposed to a read speech) has been collected to retain the prosody. As natural database is very important, it has been collected and further details are given in Table 2.

### 3.2 Pre-processing

In preprocessing two tasks are considered: one is silence removal and the second one is energy valley based segmentation. The following two paragraphs describe the same in detail.

Prior to the extraction of any features, the unwanted silence portion of the utterances has to be removed as it does not contain any useful information. There are various methods proposed in the literature using Short Time Energy (STE) and Zero Crossing Rate (ZCR) (Donald et al. 1989; Young et al. 1997). The noise is a signal which looks like whisper sounds of a phoneme set. It is added to the signal due to surrounding sound even can be found even in studio recordings. The traditional energy and zero crossing rate (ZCR) may not give accurate results in such cases. A novel approach has proposed in this work for silence removal. The features such as MFCCs have been extracted from consecutive frames of length 25 ms and feeded as an input to a two layer neural network, trained to classify each window as silent or non-silent. The outcome of silence removal is depicted in Fig. 2. Silent frames of the Fig. 2 a are discarded and non-silent frames are concatenated in order (as shown in Fig. 2 b). An accuracy of 98.3% in identifying non-silence frames from the given speech signal which successfully separating all the silence regions compared to traditional approaches. The detailed process of silence removal is explained in Algorithm 1. In general, silence can be removed from a speech by applying a threshold to energy. It is also true that the whisper sound and noise in the silence portion have the same energy. The thresholding based techniques may not be useful in such cases. It is found that the features named as Mel frequency cepstral coefficients (MFCCs) carry different information for silence and non-silence portions. Hence, we extracted the MFCCs from silence and non-silence portions and considered them as two classes. The artificial neural networks (ANNs) have been used as classifiers. This approach is found to be better in all the cases to segment silence and non-silence portions when compared to the traditional approach. The silence portion recognition rate is around 98.3% with this approach whereas,

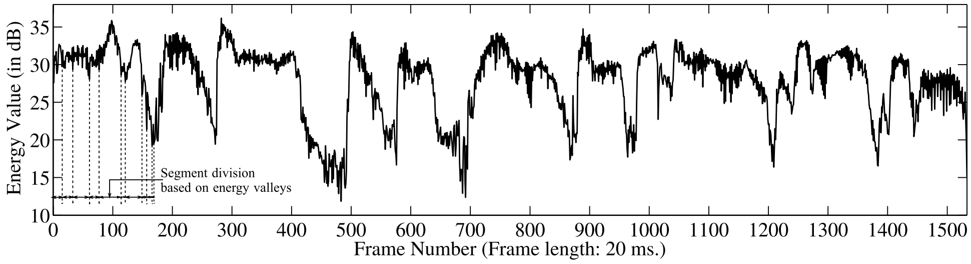

**Fig. 3** Diagrammatic representation of energy valley segmentation

the traditional approach is not even crossing 70%. The basic network structure and the process of MFCC extraction are inherited and modified accordingly to make them suitable for this work.

```
ALGORITHM 1: Silence Removal Algorithm
Input: Speech signal with silent and non-silent regions.
Output: Speech signal with non-silent regions.
$fs \leftarrow 16k$; $Tw \leftarrow 25$; $Ts \leftarrow 12.5$;
$Nw \leftarrow round(10^{-3} * Tw * fs)$;
$Ns \leftarrow round(10^{-3} * Ts * fs)$;
// $Nw \leftarrow$ the number of samples in window and $Ns \leftarrow$ the number to
  shift the samples.
for each speech sample $i$ in $1 \ldots n$ do
    $speech$ = wavread('$sample(i).wav$');
    $segment[]$ = vec2frames($speech, Nw, Ns$);
    // It will convert the speech signal into 25ms segments with 10ms
      overlap.
    for each segment $j$ in $1 \ldots segment.length$ do
        $inputs[i][j]$ = extractMFCC($segment[j]$));
    end
end
$labels[][]$ = [silent; non-silent];
train($net, inputs, labels$);
$test\_sample$ = wavread('input.wav');
$segment[]$ = vec2frames($test\_sample, Nw, Ns$);
$speech\_signal$ = null;
for each segment $i$ in $1 \ldots segment.length$ do
    $mfcc[][]$ = extractMFCC($segment(i)$);
    $label$ = test(net,mfcc);
    if $label$ == non-silent then
        $speech\_signal$ += $segment[i]$;
    end
    else
        No operation
    end
end
```

The second task is segmentation based on energy valleys', which is considered to extract features from variable length frames rather conventional fixed length frames. Each of the features mentioned in this work must be observed over a certain portion of the recorded speech sample. This portion is typically much smaller than the full speech length. In this work, the energy valley based segmentation procedure involves finding the energy minima separated (at least) by a minimum length interval (manually set) (Ellis

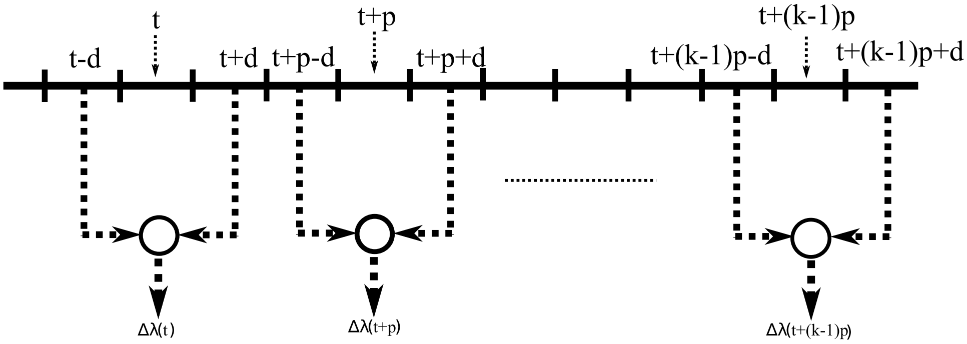

**Fig. 4** SDC feature extraction with parameters (*n*-*p*-*d*-*k*)

2005). It is desired that each window captures the vowel part of a speech sample, however, prolonged or protracted that may be. Fixing the window size would involve making an assumption about the approximate speaking rate, which defeats this purpose. The portion of speech between two consecutive energy minima forms a segment using energy valleys as the de-limitation points for segment lengths (as shown in Fig. 3).

## 3.3 Feature extraction

In this experiment the combination of spectral features such as MFCC and SDC along with prosodic features have been considered. This subsection explains the reason and stepwise process of extracting them.

### 3.3.1 *MFCC features*

This work uses an MFCC extraction algorithm based on Rastamat routines explained in Huang et al. (2001). Since the MFCCs are highly correlating human perception process, they have been considered as baseline features for many speech tasks. The Rasta features have been considered to retain the information even in noise conditions. The segment obtained by energy valleys' based segmentation is further divided into 25 ms frames with an overlap of 10 ms. The average of the features obtained for that segment has been considered as segment level features. For each segment, the feature dimension length is 13.

### 3.3.2 *SDC features*

These are the variant features of MFCC stacking cepstra Buttkus (2000) and $\Delta$ cepstral (Kumar et al. 2011) features which are computed at different speech frames to construct shifted delta cepstral (SDC) features. Four parameters ($n, p, d, k$) are required to define SDC where the total number of cepstral coefficients are denoted with $n$, $p$ is the time shift, $d$ is the time advance and span of feature is determined by $k$. For a given utterance with $n_v$ number of cepstral features, $[n_v - (k-1) \times p - d]$ number of SDC features can be extracted. The process of SDC feature extraction is shown in Fig. 4. SDC feature ($\lambda$) at the time $t$ is given as:

$$\Delta\lambda(t) = \lambda(t + jp + d) - \lambda(t + jp - d) \tag{1}$$

where $j$ lies between 0 to $k-1$.

### 3.3.3 *Prosodic features*

The prosodic features namely pitch contour, energy contour, zero crossing rate (ZCR), and duration between consecutive energy valleys in the speech signal have been considered (to capture the rate of speech). Pitch contour captures the characteristics that are pertaining to articulation. Energy contour captures stress patterns in speech. The energy valleys in speech serve as delimiters for phonemes or vowels in speech (Sreenivasa Rao and Nandi 2015). Thus, the duration between two energy valleys serves as a correlate to estimate the speaking rate. This subsection details the process that has been considered to extract the prosodic features which is quiet different from other existing approaches.

3.3.3.1 *Pitch contour* Initially, the fundamental frequency (F0) or pitch[1] is computed from each segment. From each segment, the pitch value is computed to draw the contour, also known as pitch contour. The concept of auto-correlation method is used to obtain the F0 value (Loizou 1998). On an average for each segment, about 500 F0 values are obtained. Further, a Legendre polynomial is fit to these values. Legendre polynomials of order 16 have been used while fitting to the obtained F0 values. The decision to use the polynomials with least number of coefficients that on an average gives a good fit, for about 500 samples. The coefficients of these 16 Legendre polynomials form a 16-dimensional feature vector.

3.3.3.2 *Energy contour* The energy contour is obtained by tracking the variation in amplitude of the signal over time. The group of multiple consecutive amplitudes together considered to obtain the Root Mean Square (RMS) value (i.e.,

[1] The terms 'pitch' and 'F0' are interchangeably used in the article.

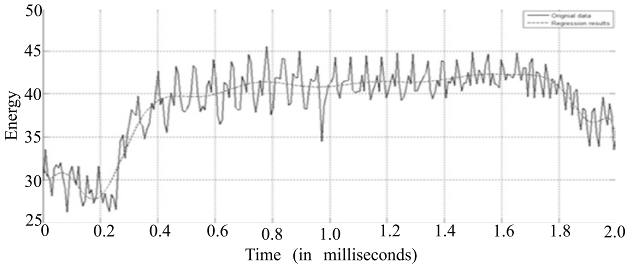

**Fig. 5** Energy contour fit (line shows the Legendre fit)

*energy*) instead of using raw amplitude values. Doing so yields about 500 RMS values per segment. Order 16 Legendre Polynomials are fit to this (as seen in Fig 5), to approximate the energy contour. The 16 coefficients thus obtained are appended to the prosody feature vector.

*3.3.3.3 Zero-crossing rate contour* Zero crossing rate (ZCR) is the rate of sign changes along a signal. The procedure involves finding the number of sign changes in each small window (based on the down-sampling) of the prosody segment. A contour is obtained for the variation of these values over the entire prosody segment. This contributes another 16 Legendre polynomials for the feature vector which form a 48-dimensional feature vector with three features.

*3.3.3.4 Duration* In order to capture the rate of speech, the duration of the segment under consideration is also appended to the prosodic feature vector.

At the end of the feature extraction process two different feature vectors have been obtained. The spectral feature vector contains 20 values (13−MFCC + 7−SDC). The prosodic feature vector contains 49 values (16−pitch + 16−energy + 16−ZCR + 1−duration). Further PCA is applied on each prosodic feature set that gives a prominent feature values. In this work first ten prominent value are considered after a thorough analysis.

### 3.4 ANN classifier

As the speech data is non-linear in nature an efficient tool is needed to map the non-linearity. Gaussian mixture model (GMM) is labelled as state-of-art classifier for most of the speaker recognition applications. From literature (Torres-Carrasquillo et al. 2002), it is observed that GMM is capable in identifying the language and also helps to tokenize the phonemes. GMM with universal background model [UBM] is also used to model the language using maximum a posteriori (MAP) estimation. However, GMM always assumes the data in normal distribution which is always not proper and GMMs are parametric estimation models. Hence, an attempt has been made in this work by using artificial neural networks (ANNs) for language identification. ANN is assumed to be effective to map the non-linear data. Two ANNs are used here: one is to classify based on spectral features and the other is for prosodic features. The structure of ANNs is shown in Fig. 6.

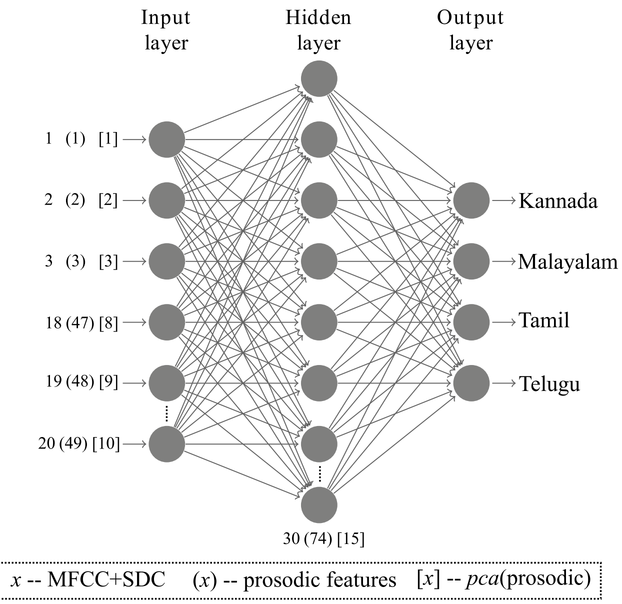

**Fig. 6** ANN classifier

Feed forward back propagation neural network (BPNN) is considered with one input, hidden and output layers. The number of neurons in input layer is equal to the size of input feature vector. The number of hidden neurons is equal to the 1.5 times than that of the number of input neurons (Gnana and Deepa 2013). Output layer contains four neurons for four language classes. Based on weighted probability ANN labels the target language class. The classification procedure is explained in the following subsection.

#### *3.4.1 Classification procedure*

Figure 7 represents in simple terms, the proposed classification procedure. We use a system of two classifiers, one that uses spectral features and the other one uses prosodic features that are obtained from each energy valley based segments. Each of the classifiers gives a class label based on the input features it receives. Best of the two approaches are considered to combine the two discrete outputs where if either of the classifiers classifies the speech sample correctly then it is treated as the correctly classified speech sample. This can be approximated by an evidence based, weighted voting system (Dietterich 2000), where each classifier output

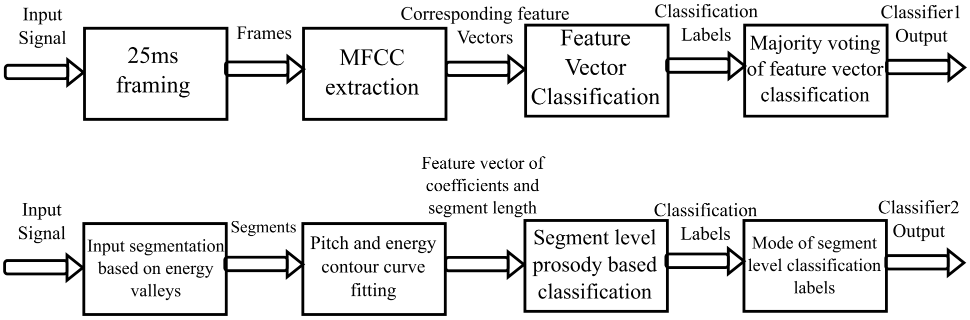
Input Signal
25ms framing
Frames
MFCC extraction
Corresponding feature Vectors
Feature Vector Classification
Classification Labels
Majority voting of feature vector classification
Classifier1 Output
Input Signal
Input segmentation based on energy valleys
Segments
Pitch and energy contour curve fitting
Feature vector of coefficients and segment length
Segment level prosody based classification
Classification Labels
Mode of segment level classification labels
Classifier2 Output

**Fig. 7** Flowchart represents classification procedure for both the spectral and prosodic features

**Table 3** Confusion matrix for language classification using MFCC features

| Target/accuracy (%) | Kannada | Malayalam | Tamil | Telugu |
|---|---|---|---|---|
| Kannada | **73.60** | 6.10 | 7.90 | 12.90 |
| Malayalam | 4.10 | **72.00** | 16.20 | 7.80 |
| Tamil | 5.30 | 19.70 | **65.10** | 9.80 |
| Telugu | 8.80 | 8.10 | 19.70 | **68.80** |

Bold values indicate the identification rate of a given class

receives a different weight for each class label it produces. The weight is simply the validation/test set accuracy of that classifier, for that class. The predicted results are represented with the support of confusion matrix.

## 4 Result analysis

Initially, a baseline system that uses MFCC features for classification is developed with a neural network classifier. For the results quoted in Table 3, the IIIT-INDIC database is used. This system was evaluated at an utterance level and the the confusion matrix obtained is shown in Table 3. The process of feature extraction and classification procedure is elaborated in Algorithm 2 for better understanding.

The neural network was trained using a training set of about 50,000 feature vectors. For validation, a separate set of about 20,000 feature vectors is used. Note that this set was completely different and mutually exclusive with the training set and training is continued till the validation error value reduced to below a threshold (0.1, averaged across all training examples).

It is found that accuracy is a little lesser on the validation data set, as expected. However overall, the accuracy averages out to about 70% at the feature level. It is worth noting that Kannada was identified with much better accuracy than the rest (86% as opposed to 57.9% for Telugu). This is probably due to the fact that the Kannada speakers voice is with little higher pitch than the others, and this finds reflection in the features we have used. In the following experiments, pitch normalization has been performed to reduce the discrepancy due to pitch variation.

**ALGORITHM 2:** Dravidian Language Classification System

```
Input: Speech Samples
Output: Regional language information to which the given speech
        belongs to
silenceRemoval(); // Ref. Algorithm 1
for each sample $i$ in $1 \ldots n$ do
    $speech = wavread('sample(i).wav')$;
    $segment[] = vec2frames(speech, Nw, Ns)$;
    for each segment $j$ in $1 \ldots segment.length$ do
        $mfcc[][] +=$ extractMFCC(segment[j]);
        $sdc[][] +=$ computeSDC(mfcc[j],segment[j]);
    end
    spectral[][] = [mfcc;sdc];
    $segment1[] =$ energyValleySeg(speech);
    // Signal into segments based on minimum energy values.
    for each segment $j$ in $1 \ldots segment1.length$ do
        $F0[] =$ extractF0(segment1[j]); // F0 values extraction.
        $pitch[] =$ legendrePolyFit(F0); // Legendre poly. fit on F0.
        $energy[] =$ extractEnergy(segment1[j]); // RMS values
        $energy1[] =$ legendrePolyFit(energy); // Legendre poly. fit on E.
        $zcr[] =$ extractZCR(segment1[j]); // ZCR values.
        $zcr1[] =$ legendrePolyFit(zcr); // Legendre poly. fit on Z.
        $duration =$ extractDuration(segment1[j]); // Speech rate.
    end
    $feature\_vector[] +=$ pitch+energy+zcr1+duration;
end
$labels[][] =$ [Kannada; Malayalam; Tamil; Telugu];
train1($net, mfcc, labels$); train2($net, feature\_vector, labels$);
$test\_sample[] =$ wavread('input.wav');
$segment[] =$ vec2frames($test\_sample, Nw, Ns$);
 $mfcc[][] =$ extractMFCC(test_sample);
$featureVector1[] =$ extractFeatures();
$output1 =$ test1(net,spectral); $output2 =$ test2(net,featureVector1);
$class\_label =$ classify(output1, output2);
// This function classifies based on the best of two outputs.
```

Even more interesting observation is that there seem to be two pairs of languages that have maximum confusion between each other. Kannada seems to be mistaken for Telugu (8.8%) more often than Tamil (5.3%) or Malayalam (4.1%). This also hold true for Telugu, where the confusion with Kannada (12.5%) is even more clear as opposed to Tamil (9.8%) and Malayalam (7.8%). The reason for this confusion pair could be that Kannada and Telugu languages share the same script and as such, have similar sounds. Since MFCC features essentially reflect how humans produce and perceive speech, the fact that Kannada and Telugu share the same script (and thus the same phoneme set) might lead to mutual confusion.

Similarly, it is found that Malayalam is mistaken for Tamil (19.7%) much often than Kannada (6.1%) or Telugu (8.1%). Same holds for Tamil, which was mistaken for Malayalam (16.2%) more than Kannada (7.9%) or Telugu (14.4%). Based on the observations, both Tamil and Telugu show markedly different pronunciations of certain letters such as L. The MFCC features could be picking up this dissimilarity, which is being manifested in the form of high confusion between the two.

At the word level, a majority voting based approach is used to make a decision as to which language the entire word belongs to. However, the threshold that qualifies as a majority must be empirically defined. Based on our experiments, it is found that the variation of accuracy in classification with respect to the threshold is related to the recall at the feature vector level for each class. The results are tabulated in Table 4. The drop in classification accuracy on

**Table 4** MFCC based classification accuracy at different thresholds

| Majority threshold (%) | Classification accuracy (%) |
|---|---|
| 50 | 100 |
| 60 | 93 |
| 70 | 75 |

increasing the threshold from 50 to 60% is probably due to Telugu examples getting misclassified since the recall for Telugu is 57% which is a little below 60%. The significant drop on increasing the threshold further, to 70%, is due to both Tamil and Malayalam not winning the majority vote, since their recall percentages are 66.5% (Tamil) and 65.6% (Malayalam) respectively.

All the experiments presented until now, have been performed on what could be argued to be an artificial data set, consisting of read speech. Moreover, it was recorded in a studio setting and is thus of very high quality. If a data set consisting of natural speech as in radio or TV chat shows is used, the results degrade even further. In fact, even with a threshold of just 50% to win a majority vote, classification accuracy is 68.3% on the radio data set, as opposed to 100% on the IIIT-Hyderabad data set. Prosodic features were tried next. Firstly, the hypothesis that the IIIT-Hyderabad INDIC data set is tested, which consists of read speech in monotonous tone would not yield a good classification accuracy based on prosodic features, as against the radio data set consisting of natural speech. Table 5 summarizes the results.

The hypothesis made earlier was found to be correct. Therefore, in our further experiments, the radio data set was used since the aim of this project is to improve classification accuracy for language identification, on natural speech using prosodic features. As explained in the Sect. 3, prosodic feature vectors have 49 features. This leads to long training and testing time. In an effort to cut down on both of these, Principal Component Analysis (PCA) was used to find a lower dimensional representation of the original feature vectors, that could hopefully, provide the same performance. The original 49 features were projected onto a 30 dimensional feature space and all experiments were repeated. An average accuracy of 12.50% on IIITH-INDIC data set and 41.50% on radio data set is achieved with prosodic features alone. The reduced dimensional set is used for this experimentation and it is further processed on the combinational feature set with two classifiers. One set contains MFCCs and SDCs while the other set contains prosodic features with PCA and without PCA. Since the accuracy is a performance of the entire system, overall accuracy is computed at every equal error rate (EER) instead of $C_{avg}$ (Nanavati 2002).

The audio clips are divided into 5, 2.5 and 1.25 s. This is done to compare the performance of the system with shorter clips. If it is better at shorter clips then automatically the processing speed of the system increases. In this case, no need to analyze the lengthy audio clips. A small portion of 1~2 s is sufficient to determine the language. The results of various combinational features are given in Table. 5. The table contains mainly three columns: first one represents feature set considered, the second one shows the results before applying the technique of PCA and the third one represents the results after applying it. The same results are displayed if the feature set is spectral alone (e.g.: see row-1). This is because PCA is not applied on spectral features. It is seen that application of PCA not only results in shorter feature

**Table 5** The performance obtained for Dravidian language identification using different combination of features

| Method | Before PCA (%) | | | After PCA (%) | | |
|---|---|---|---|---|---|---|
| | 5 s input samples | 2.5 s input samples | 1.25 s input samples | 5 s input samples | 2.5 s input samples | 1.25 s input samples |
| MFCCs [13,13]* | 68.30 | 68.30 | 62.50 | 68.30 | 68.30 | 62.50 |
| MFCCs+SDCs [20,20] | 72.90 | 71.40 | 66.30 | 72.90 | 71.40 | 66.30 |
| P+E [32,20] | 47.50 | 32.50 | 31.90 | 47.50 | 36.30 | 37.50 |
| E+Z [32,20] | 45.00 | 40.00 | 34.40 | 25.00 | 22.50 | 26.30 |
| Z+P [32,20] | 37.50 | 30.00 | 25.60 | 25.00 | 25.00 | 25.00 |
| MFCC+P+E [45,33] | 82.50 | 77.50 | 71.30 | 80.00 | 78.80 | 72.00 |
| MFCC+SDC+ P+E [52,40] | **86.70** | 83.40 | 78.50 | **81.30** | 80.20 | 73.60 |
| MFCC+E+Z [45,33] | 72.50 | 73.80 | 68.10 | 70.00 | 70.00 | 62.50 |
| MFCC+SDC+ E+Z [52,40] | 77.00 | 73.20 | 71.80 | 72.40 | 71.90 | 66.10 |
| MFCC+Z+P [45,33] | 80.00 | 77.50 | 68.80 | 70.00 | 70.00 | 62.50 |
| MFCC+SDC+ Z+P [52,40] | 82.60 | 80.40 | 71.30 | 76.80 | 74.30 | 69.70 |

*[$x$, $y$] indicates the length of feature vector before and after PCA

*P* Pitch contour fit based on auto-correlation, *E* energy contour fit, *Z* zero crossing rate and bold font indicates better results vectors but as in the case of "MFCC+SDC+P+E" we see an improvement in identification performance for the shorter input samples of length 2.5 and 1.25 s. This is especially valuable since greater success on shorter clips increases feasibility for a real-time system as maybe required by many applications.

## 5 Conclusion and future work

The present work investigates the viability of prosodic features for the task of naturally spoken language identification of Dravidian language set. To reduce the computational complexity Legendre polynomial fitting is done and for further reduction, PCA is applied to construct a low dimensional feature vector. The silence removal is done using ANNs that gave better performance when compared to traditional approaches. The segmentation is done based on energy valleys' instead of fixed length frames. Different combinations are considered to combine these features with MFCCs and SDCs to compare and analyze the performance of the system. ANNs are considered to classify the two sets of four languages.

While the results are encouraging, there is scope for improvement. As future work, a vowel specific and syllable specific analysis and classification can yield valuable insights, especially for Dravidian languages where it is known that these pronunciations vary across languages. Recent developments in the task of automated vowel onset point detection gives a language independent technique for segmenting out the vowels present in a utterance. This opens up the possibility of near phonatactic performance without the processing overhead or language specificity involved. This possibility holds exciting prospects. Multi-level classification also helps to recognize the various sets initially, and further, to recognize the language.

## References

Allen, F., Ambikairajah, E., & Epps, J. (2005). Language identification using warping and the shifted delta cepstrum. In *IEEE 7th workshop on multimedia signal processing*, pp. 1–4. IEEE.

Atal, B., & Rabiner, L. (1946). pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *24*(3), 201–212.

Brümmer, N., Cumani, S., Glembek, O., Karafiát, M., Matějka, P., Pešán, J., Plchot, O., Soufifar, M., Villiers, E. D., & Cernockỳ, J. H. (2012). Description and analysis of the brno276 system for lre2011. In *Odyssey 2012-the speaker and language recognition workshop*.

Buttkus, B. (2000). *Spectral Analysis and Filter Theory in Applied Geophysics: With 23 Tables*. Berlin: Springer Science & Business Media.

Chandrasekaran, K. (2012). Indeterminacies in howatch's st. benet's trilogy. *Language in India*, 12(12).

Childers, D. G., Hahn, M., & Larar, J. N. (1989). Silent and voiced/unvoiced/mixed excitation (four-way) classification of speech. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *37*(11), 1771–1774.

Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep big simple neural nets excel on handwritten digit recognition []. Retrieved July 03, 2014, from: http://arxiv.orgpdf/1003.0358.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning*, pp. 160–167. ACM.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., & Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(4), 788–798.

Dehak, N., Torres-Carrasquillo, P. A., Reynolds, D., & Dehak, R. (2011). Language recognition via i-vectors and dimensionality reduction. In *Twelfth annual conference of the international speech communication association*.

Deng, L., Dong, Y., et al. (2014). Deep learning: Methods and applications. Foundations and Trends®. *Signal Processing*, *7*(3–4), 197–387.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple classifier systems*, pp. 1–15. Springer.

Ellis, D. (2005). Reproducing the feature outputs of common programs using matlab and melfcc.

Ganapathy, S., Han, K., Thomas, S., Omar, M., Segbroeck, M. V., & Narayanan, S. S. (2014). Robust language identification using convolutional neural network features. In *Fifteenth annual conference of the international speech communication association*.

Gnana S. K., & Deepa, S. N. (2013). Review on methods to fix number of hidden neurons in neural networks. In *Mathematical Problems in Engineering*.

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 6645–6649. IEEE.

Hinton, G., Deng, L., Dong, Y., Dahl, G. E., Mohamed, A. R., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97.

Huang, X., Acero, A., Hon, H. W., & Reddy, R. (2001). *Spoken language processing: A guide to theory, algorithm, and system development*. Upper Saddle River: Prentice Hall PTR.

Jain, D., & Cardona, G. (2007). *The Indo-Aryan Languages*. Abingdon: Routledge.

Jiang, B., Song, Y., Wei, S., McLoughlin, I. V., & Dai, L. R. (2014). Task-aware deep bottleneck features for spoken language identification. In *Proceedings of the 15th annual conference of the international speech communication association (INTERSPECH), Singapore*.

Kumar, K., Kim, C., & Stern, R. M. (2011). Delta-spectral cepstral coefficients for robust speech recognition. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4784–4787. IEEE.

Li, H., Ma, B., & Lee, K. A. (2013). Spoken language recognition: From fundamentals to practice. *Proceedings of the IEEE*, *101*(5), 1136–1159.

Li, H., & Ma, B. (2005). A phonotactic language model for spoken language identification. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pp. 515–522. Association for Computational Linguistics.

Loizou, P. (1998). *A matlab software tool for speech analysis*. Dallas: Author.

Lopez-Moreno, I., Gonzalez-Dominguez, J., Martinez, D., Plchot, O., Gonzalez-Rodriguez, J., & Moreno, P. J. (2016). On the use of deep feedforward neural networks for automatic language identification. *Computer Speech and Language*, *40*, 46–59.

Lopez-Moreno, I., Gonzalez-Dominguez, J., Plchot, O., Martinez, D., Gonzalez-Rodriguez, J., & Moreno, P. (2014). Automatic language identification using deep neural networks. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5337–5341. IEEE.

Martínez, D., Burget, L., Ferrer, L., & Scheffer, N. (2012). ivector-based prosodic system for language identification. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4861–4864. IEEE.

Matejka, P., Burget, L., Schwarz, P., & Cernocky, J. (2006). Brno university of technology system for nist 2005 language recognition evaluation. In *The IEEE Odyssey speaker and language recognition workshop*, pp. 1–7. IEEE.

Matejka, P., Schwarz, P., Cernockỳ, J., & Chytil, P. (2005). Phonotactic language identification using high quality phoneme recognition. In *Interspeech*, pp. 2237–2240.

Mohamed, A. R., Dahl, G. E., & Hinton, G. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(1), 14–22.

Montavon, G. (2009). Deep learning for spoken language identification. In *NIPS workshop on deep learning for speech recognition and related applications*.

Nanavati, T. (2002). *Biometrics*. New York: Wiley.

Ng, R.W., Leung, C.C., Lee, T., Ma, B., & Li, H. (2010). Prosodic attribute model for spoken language identification. In *IEEE international conference on acoustics speech and signal processing (ICASSP)*, pp. 5022–5025. IEEE.

Pinto, J., Yegnanarayana, B., Hermansky, H., & Doss, M. M. (2008). Exploiting contextual information for improved phoneme recognition. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4449–4452. IEEE.

Prahallad, K., Kumar E. N., Keri V., Rajendran, S., & Black, A. W. (2012). In *INTERSPEECH* TheIIIT-HIndic speech databases.

Ranjan, S., Yu, C., Zhang, C., Kelly, F., & Hansen, J. H. (2016). Language recognition using deep neural networks with very limited training data. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5830–5834. IEEE.

Rao, K. S., & Nandi, D. (2015). *Language Identification Using Excitation Source Features*. Berlin: Springer.

Singer, E., Torres-Carrasquillo, P., Reynolds, D. A., McCree, A., Richardson, F., Dehak, N., & Sturim, D. (2012). The mitll nist lre 2011 language recognition system. In *IEEE international conference on acoustics speech and signal processing (ICASSP)*, pp. 209–215.

Sturim, D., Campbell, W., Dehak, N., Karam, Z., McCree, A., Reynolds, D., Richardson, F., Torres-Carrasquillo, P., & Shum, S. (2011). The mit ll 2010 speaker recognition evaluation system: Scalable language-independent speaker recognition. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5272–5275. IEEE.

Torres-Carrasquillo, P. A., Reynolds, D., & Deller, J. R. Jr. (2002). Language identification usingGaussian mixture model tokenization. In *IEEE international conference on acoustics, speech, and signal processing (ICASSP)* (Vol. 1, pp. I–757). IEEE.

Torres-Carrasquillo, P. A., Singer, E., Kohler, M. A., Greene, R. J., Reynolds, D. A., & Deller Jr., J. R. (2002). Approaches to language identification using Gaussian mixture models and shifted delta cepstral features. In *Interspeech*.

Torres-Carrasquillo P. A., Singer E., Gleason T., McCree A., Reynolds D. A., Richardson F., & Sturim, D. (2010). The mitll nist lre 2009 language recognition system. In *IEEE international conference on acoustics speech and signal processing (ICASSP)*, pp. 4994–4997. IEEE.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., & Povey, D. (1997). In *The HTK book* (Vol. 2. Entropic Cambridge Research Laboratory Cambridge).

Zissman, M. A. (1996). Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on Speech and Audio Processing*, *4*(1), 31.

Zissman, M. A. (1995). Language identification using phoneme recognition and phonotactic language modeling. In *International conference on acoustics, speech, and signal processing (ICASSP)* (Vol. 5, pp. 3503–3506). IEEE.