Published in IET Signal Processing
Received on 21st May 2012
Revised on 26th November 2012
Accepted on 8th January 2013
doi: 10.1049/iet-spr.2012.0151

ISSN 1751-9675

Comparative study of automatic speech recognition techniques

Michelle Cutajar, Edward Gatt, Ivan Grech, Owen Casha, Joseph Micallef

Faculty of Information and Communication Technology, Department of Microelectronics and Nanoelectronics, University of Malta, Tal-Qroqq, Msida, MSD 2080, Malta
E-mail: mcut0007@um.edu.mt

Abstract: Over the past decades, extensive research has been carried out on various possible implementations of automatic speech recognition (ASR) systems. The most renowned algorithms in the field of ASR are the mel-frequency cepstral coefficients and the hidden Markov models. However, there are also other methods, such as wavelet-based transforms, artificial neural networks and support vector machines, which are becoming more popular. This review article presents a comparative study on different approaches that were proposed for the task of ASR, and which are widely used nowadays.

1 Introduction

Human beings find it easier to communicate and express their ideas via speech. In fact, using speech as a means of controlling one's surroundings has always been an intriguing concept. For this reason, automatic speech recognition (ASR) has always been a renowned area of research. Over the past decades, a lot of research has been carried out in order to create the ideal system which is able to understand continuous speech in real-time, from different speakers and in any environment. However, the present ASR systems are still far from reaching this ultimate goal [1, 2].

Large variations in speech signals make this task even more challenging. As a matter of fact, even if the same phrase is pronounced by the same speaker for a number of times, the resultant speech signals will still have some small differences. A number of difficulties that are encountered during the recognition of speech signals are the absence of clear boundaries between phonemes or words, unwanted noise signals from the speaker's surrounding environment and speaker variability, such as gender, speaking style, speed of speech, and regional and social dialects [3, 4].

Various applications where ASR is, or can be employed, vary from simple tasks to more complex ones. Some of these are speech-to-text input, ticket reservations, air traffic control, security and biometric identification, gaming, home automation and automobile sectors [5, 6]. In addition, disabled and elderly persons can highly benefit from advances in the field of ASR.

Over the past years, several review papers were published, in which the ASR task was examined from various perspectives. A recent review [7] discussed some of the ASR challenges and also presented a brief overview on a number of well-known approaches. The authors considered two feature extraction techniques: the linear predictive coding coefficient (LPCC) and the mel frequency cepstral coefficient (MFCC), as well as five different classification methods: template-based approaches, knowledge-based approaches, artificial neural networks (ANNs), dynamic time warping (DTW) and hidden Markov models (HMMs). Finally, a number of ASR systems were compared, based on the feature extraction and classification techniques used. Another review paper [8] presented the numerous possible digital representations of a speech signal. Hence, the authors focused on numerous approaches that were employed at the pre-processing and feature extraction stages of an ASR system. A different viewpoint on the construction of ASR systems is presented in [9], where the author points out that an ASR system consists of a number of processing layers, since several components are required, resulting in a number of computational layers. The author also states that the present error rates of ASR systems can be reduced, if the corresponding processing layers are chosen wisely. Another two important review papers, written by the same author, are presented in [4, 10]. In [10], the author discusses both ASR and text-to-speech (TTS) research areas. Considering only the ASR section, different aspects were considered, such as data compression, cepstrum-based feature extraction techniques and HMMs for the classification of speech. In addition, different ways to increase robustness against noise, were also discussed. As for the review paper presented in [4], the field of ASR is discussed from the viewpoint of pattern recognition. Different problems that are encountered and various methods on how to perform pattern recognition of speech signals are discussed. These methods are all discussed with respect to the nature of speech signals, in order to obtain data reduction.

In this review paper, an analysis on different techniques that are widely being employed nowadays for the task of ASR is presented. In the following sections, the basic ASR

model is introduced, along with a discussion on the various methods that can be used for the corresponding components. A comparison on different ASR systems that were proposed will be presented, along with a discussion on the progress of ASR techniques.

2 Automatic speech recognition systems

For an ASR system, a speech signal refers to the analogue electrical representation of the acoustic wave, which is a result of the constrictions in the vocal tract. Different vocal tract constrictions generate different sounds. Most ASR systems take advantage of the fact that the change in vocal tract constrictions between one sound and another is not done instantly. Hence, for a small portion of time, the vocal tract is stationary for each sound, and this is usually taken to be between 10 and 20 ms. The basic sound in a speech signal is called a phoneme. These phonemes are then combined, to form words and sentences. Each phoneme is dependent on its context, and this dependency is usually tackled, by considering tri-phones. Each language has its own set of distinctive phonemes, which typically amounts to between 30 and 50 phonemes. For example, the English language can be represented by approximately 42 phonemes [3, 8, 11, 12].

An ASR system mainly consists of four components: pre-processing stage, feature extraction stage, classification stage and a language model, as shown in Fig. 1. The pre-processing stage transforms the speech signal before any information is extracted by the feature extraction stage. As a matter of fact, the functions to be implemented by the pre-processing stage are also dependent on the approach that will be employed at the feature extraction stage. A number of common functions are the noise removal, endpoint detection, pre-emphasis, framing and normalisation [10, 13, 14].

After pre-processing, the feature extraction stage extracts a number of predefined features from the processed speech signal. These extracted features must be able to discriminate between classes while being robust to any external conditions, such as noise. Therefore, the performance of the ASR system is highly dependent on the feature extraction method chosen, since the classification stage will have to classify efficiently the input speech signal according to these extracted features [15–17]. Over the past few years various feature extraction methods have been proposed, namely the MFCCs, the discrete wavelet transforms (DWTs) and the linear predictive coding (LPC) [1, 5].

The next stage is the language model, which consists of various kinds of knowledge related to a language, such as the syntax and the semantics [18]. A language model is required, when it is necessary to recognise not only the phonemes that make up the input speech signal, but also in moving to either trigram, words or even sentences. Thus, knowledge of a language is necessary in order to produce meaningful representations of the speech signal [19].

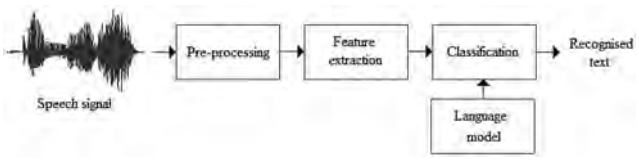

Fig. 1 *Traditional ASR system [10, 13]*

The final component is the classification stage, where the extracted features and the language model are used to recognise the speech signal. The classification stage can be tackled in two different ways. The first approach is the generative approach, where the joint probability distribution is found over the given observations and the class labels. The resulting joint probability distribution is then used to predict the output for a new input. Two popular methods that are based on the generative approach are the HMMs and the Gaussian mixture models (GMMs). The second approach is called the discriminative approach. A model based on a discriminative approach finds the conditional distribution using a parametric model, where the parameters are determined from a training set consisting of pairs of the input vectors and their corresponding target output vectors. Two popular methods that are based on the discriminative approach are the ANNs and support vector machines (SVMs) [20, 21]. Various researches focused on using only one method for the classification stage, such as the HMMs, which is the mostly used method in the field of ASR. However, numerous ASR systems based on hybrid models were also proposed, in order to combine the strengths of both approaches.

In the following sections, various methods that were proposed for the feature extraction stage, the classification stage, and the language model are going to be discussed into further detail, with special reference to those algorithms that are widely used nowadays.

2.1 Feature extraction stage

The mostly renowned feature extraction method in the field of ASR is the MFCC. However, apart from this technique, there are also other feature extraction methods, such as the DWT and the LPC, which are also highly effective for ASR applications.

2.1.1 Mel-frequency cepstral coefficients: Numerous researchers chose MFCC as their feature extraction method [22–26]. As a matter of fact, since the mid-1980s, MFCCs are the most widely used feature extraction method in the field of ASR [10, 27].

The MFCC try to mimic the human ear, where frequencies are nonlinearly resolved across the audio spectrum. Hence, the purpose of the mel filters is to deform the frequency such that it follows the spatial relationship of the hair cell distribution of the human ear. Hence, the mel frequency scale corresponds to a linear scale below 1 kHz, and a logarithmic scale above the 1 kHz, as given by (1) [28, 29].

$$F_{\text{mel}} = \frac{1000}{\log(2)} \times \left[1 + \frac{F_{\text{Hz}}}{1000}\right] \tag{1}$$

The computation of the MFCC is carried out by first dividing the speech signal into overlapping frames of duration 25 ms [22, 25, 26] or 30 ms [2, 28], with 10 ms of overlap for consecutive frames. Each frame is then multiplied with a Hamming window function, and the discrete Fourier transform (DFT) is computed on each windowed frame [13, 28]. Generally, instead of the DFT, the fast Fourier transform (FFT) is adopted to minimise the required computations [10]. Subsequently, the data obtained from the FFT are converted into filter bank outputs and the log energy output is evaluated, as shown in (2), where $H_i(k)$ is

doi: 10.1049/iet-spr.2012.0151

the filter bank

$$X_i = \log_{10}\left(\sum_{k=0}^{N-1} |X(k)| \times H_i(k)\right), \quad \text{for } i = 1, ..., M \quad (2)$$

Finally, the direct cosine transform (DCT), shown in (3) is performed on the log energy output and the MFCC are obtained at the output. Since the DCT packs the energy into few coefficients and discards higher-order coefficients with small energy, dimensionality reduction is achieved while preserving most of the energy [13, 28]

$$C_j = \sum_{i=1}^{M} X_i \cos\left(j \times \left(i - \frac{1}{2}\right) \times \frac{\pi}{M}\right), \quad \text{for } j = 0, ..., J-1 \quad (3)$$

Although for the computation of MFCC, the speech signal is divided into frames of duration 25 or 30 ms, it is important to point out that the co-articulation of a phoneme extends well beyond 30 ms. Thus, it is important to take into account also the timing correlations between frames. With MFCC this is taken into consideration by the addition of the dynamic and acceleration features, commonly known as delta and delta–delta features. Thus, the MFCC feature vector normally consists of the static features, which are obtained from the analysis of each frame, the dynamic features, namely the differences between static features of successive frames, and finally the acceleration features, which are the differences between the dynamic features. A typical MFCC feature vector consists of 13 static cepstral coefficients, 13 delta values and 13 delta–delta values, resulting in a 39-dimensional feature vector [10]. Another commonly used MFCC feature vector takes into consideration the normalised log energy. Hence, instead of having 13 static cepstral coefficients, the MFCC feature vector would consist of 12 static cepstral coefficients along with the normalised log energy, with the addition of the corresponding dynamic and acceleration features. This would result also into a 39-dimensional feature vector [22, 23, 26]. The work presented in [23] shows that the addition of the dynamic and acceleration features improves the recognition rate of the whole ASR model. In this research, continuous density HMMs (CDHMMs) were implemented for the task of speaker-independent phoneme recognition, along with the MFCC as feature extraction method. From the results obtained, it was showed that for context-independent phone modelling, an increase in accuracy of approximately 8% was achieved when the normalised log energy, dynamic and acceleration features were appended to 12 static cepstral coefficients.

Although MFCC are renowned and widely used in the area of speech recognition, these still present some limitations. MFCCs main drawback is their low robustness to noise signals, since all MFCC are altered by the noise signal if at least one frequency band is distorted [25, 27, 30–32]. Apart from this, in MFCC it is inherently assumed that a frame speech contains information of only one phoneme at a time, whereas it may be the case that in a continuous speech environment a frame speech contains information of two consecutive phonemes [27, 32].

Various techniques on how to improve the robustness of MFCC with respect to noise-corrupted speech signals have been proposed. The techniques, which are widely used, are based on the concept of normalisation of the MFCCs, in both training and testing conditions [30]. Examples of features statistics normalisation techniques are mean and variance normalisation (MVN) [30], histogram equalisation (HEQ) [30] and cepstral mean normalisation (CMN) [25, 33]. In research [30], the normalisation techniques MVN and HEQ were performed in full-band and sub-band modes. With full-band mode, the chosen normalisation technique is performed directly on the MFCCs, whereas in sub-band mode, before performing the normalisation techniques on the MFCCs, the MFCCs are first decomposed into non-uniform sub-bands with the implementation of DWT. In this case, it is possible to process individually, some or all of the sub-bands, by the normalisation technique. Finally, the feature vectors are reconstructed using the inverse DWT (IDWT). Thus, this procedure allows the possibility of processing separately those spectral bands that contain essential information in the feature vectors. The results obtained in this research confirmed that the inclusion of normalisation techniques significantly improved the accuracy of the ASR system. In fact, both full-band and sub-band implementations of the MVN and HEQ normalisation techniques obtained an increase in the accuracy, with the sub-band versions performing best. With a sub-band implementation, an increase in accuracy of approximately 17% was obtained. Furthermore, HEQ outperformed MVN in almost all signal-to-noise ratio (SNR) cases considered in this study. Another research that implemented a normalisation technique is presented in [25], where the CMN is performed on the full-band MFCC feature vectors.

Another important concern with MFCCs is that these are derived from only the power spectrum of a speech signal, ignoring the phase spectrum. However, information provided by the phase spectrum is also useful for human speech perception [24]. This issue is tackled by performing speech enhancement before the feature extraction stage. The work in [24] performs speech enhancement before the feature extraction stage of the ASR model. The speech signal enhancement stage employs the perceptual wavelet packet transform (PWPT) to decompose the input speech signal into sub-bands. De-noising with PWPT is performed by the use of a thresholding algorithm. After de-noising the wavelet coefficients obtained from the PWPT, these are reconstructed by means of the inverse PWPT (IPWPT). In this research, a modified version of the MFCCs is implemented. These are the mel-frequency product spectrum cepstral coefficients (MFPSCCs), which also consider the phase spectrum during feature extraction. The results obtained show that the performance of both MFCCs and MFPSCCs is comparable for clean speech. However, for noise-corrupted speech signals, MFPSCCs achieved higher recognition rates as the SNR decreases.

2.1.2 Discrete wavelet transform: DWTs take into consideration the temporal information that is inherent in speech signals, apart from the frequency information. Since speech signals are non-stationary in nature, the temporal information is also important for speech recognition applications [2, 16, 34]. With DWT, temporal information is obtained by re-scaling and shifting an analysing mother wavelet. In this manner, the input speech signal is analysed at different frequencies with different resolutions [16, 34]. As a matter of fact, DWTs are based on multiresolution analysis, which considers the fact that high-frequency components appear for short durations, whereas

low-frequency components appear for long durations. Hence, a narrow window is used for high frequencies and a wide window is used at low frequencies [34]. For this reason, the DWT provides an adequate model for the human auditory system, since a speech signal is analysed at decreasing frequency resolution for increasing frequencies [17].

The DWT implementation consists of dividing the speech signal under test into approximation and detail coefficients. The approximation coefficients represent the high-scale low-frequency components, whereas the detail coefficients represent the low-scale high-frequency components of the speech signal [5, 16]. The DWT can be implemented by means of a fast pyramidal algorithm consisting of multirate filterbanks, which was proposed in 1989 by Stephane G. Mallat [35]. In fact, this algorithm is known as the Mallat algorithm or Mallat-tree decomposition. This pyramidal algorithm analyses the speech signal at different frequency bands with different resolutions, by decomposing the signal into approximation and detail coefficients as shown in Fig. 2. The input speech signal is passed through a low-pass filter and a high-pass filter, and then down-sampled by 2, in order to obtain the approximation and detail coefficients, respectively [16]. Hence, the approximation and detail coefficients can be expressed by (4) and (5), respectively, where $h[n]$ and $g[n]$ represent the low-pass and high-pass filters [34]

$$y_{\text{low}}[k] = \sum_n x[n] \times h[2k - n] \tag{4}$$

$$y_{\text{high}}[k] = \sum_n x[n] \times g[2k - n] \tag{5}$$

The approximation coefficients are then further divided using the same wavelet decomposition step. This is achieved by successive high-pass and low-pass filtering of the approximation coefficients. This makes DWT a potential candidate for SR tasks, since most of the information of a speech signal lies at low frequencies. As a matter of fact, if the high-frequency components are removed from a speech signal, the sound will be different, but what was said can still be understood [16]. The work in [12] confirms this, since it was shown that better accuracy is achieved when approximation coefficients are used to generate octaves, instead of using the detail coefficients.

The DWT coefficients of the input speech signal are then obtained by concatenating the approximation and detail coefficients, starting from the last level of decomposition [36]. The number of possible decomposition levels is limited by the frame size chosen, although a number of octaves between 3 and 6 are common [12].

The low-pass and high-pass filters used for DWT must be quadrature mirror filters (QMF), as shown in (6), where L is the filter length. This ensures that the filters used are half-band filters. This QMF relationship guarantees also perfect reconstruction of the input speech signal after it has been decomposed. Orthogonal wavelets such as Haar, Daubechies and Coiflets all satisfy the QMF relationship [34]

$$g[L - 1 - n] = (-1)^n \times h[n] \tag{6}$$

The complexity of DWT is also very minimal. Considering a complexity C per input sample for the first stage, because of the sub-sampling by 2 at each stage, the next stage will end up with a complexity equal to C/2 and so on. Thus, the complexity of DWT will be less than 2C [37].

Various researches employed DWT at the feature extraction stage [1, 5, 38–41]. The work proposed in [1] used DWT to recognise spoken words for the Malayalam language. A database of 20 different words, spoken by 20 individuals was utilised. Hence, an ASR system for speaker-independent isolated word recognition was designed. With DWT at the feature extraction stage, feature vectors of element size 16 were employed. At the classification stage, an ANN, the multilayer perceptron (MLP) was used. With this approach, the accuracy reached for the Malayalam language is of 89%.

Another research that explores into more detail the DWTs for ASR is presented in [5]. In this research, the DWTs are used for the recognition of the Hindi language. Different types of wavelets were used for the DWT, to verify which wavelet type will provide the highest accuracy. The wavelets that were considered in this study are as follows:

- Daubechies wavelet of order 8 with three decomposition levels;
- Daubechies wavelet of order 8 with five decomposition levels;
- Daubechies wavelet of order 10 with five decomposition levels;
- Coiflets wavelet of order 5 with five decomposition levels;
- Discrete Meyer with five decomposition levels.

The DWT coefficients obtained, were not used directly by the classification stage, since after obtaining the DWT coefficients, the LPCCs were evaluated based on these coefficients. Afterwards, the K-mean algorithm is used to form a vector quantised (VQ) codebook. During the recognition phase, the minimum squared Euclidean distance was used to find the corresponding codeword in the VQ codebook. The results obtained showed that the Daubechies wavelet of order 8 with five decomposition levels performed best, surpassing the others by an accuracy of 6%. This was followed by the Daubechies wavelet of order 10 with five decomposition levels, the discrete Meyer wavelet, the Coiflet wavelet and finally the Daubechies wavelet of order 8 with three decomposition levels. From the results obtained, it can be concluded that the Daubechies wavelet provided the higher recognition rates when compared with other wavelets that were considered, provided that enough decomposition levels were considered.

As a matter of fact, Daubechies wavelets are the most widely used wavelets in the field of ASR applications [5, 12, 16, 24, 27, 40, 42]. These are also known as the Maxflat wavelets since their frequency responses have maximum flatness at frequencies 0 and π [16, 34]. Different orders of the Daubechies wavelet were implemented in different researches, although the wavelet of order 8 is the one which is widely used [5, 12, 24, 40, 43].

A number of research publications have also shown that DWT provide better results than the MFCC. When compared

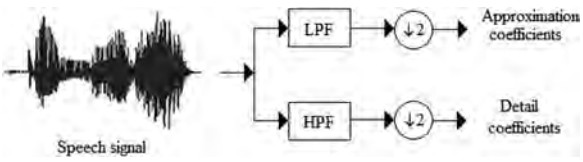

Fig. 2 *Decomposition stage [16]*

doi: 10.1049/iet-spr.2012.0151

with MFCC, the DWT enables better frequency resolution at lower frequencies, and hence better time localisation of the transient phenomena in the time domain [39, 44].

As already mentioned earlier, MFCC are not robust with respect to noise-corrupted speech signals. On the other hand, DWT were successfully used for de-noising tasks because of their ability in providing localised time and frequency information [17, 31, 45]. Hence, if only a part of the speech signal's frequency band is corrupted by noise, not all DWT coefficients are altered.

Various researchers considered the idea of merging the DWT and MFCC, in order to benefit from the advantages of both methods. This new feature extraction method is known as mel-frequency discrete wavelet coefficients (MFDWC), and is obtained by applying the DWT to the mel-scaled log filter bank energies of a speech frame [32, 41, 46]. In [46] the MFDWC method was used with DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT) database. The phonemes available in the TIMIT database were clustered to a total of 39 classes according to the CMU/MIT standards. The results obtained showed that MFDWC achieved higher accuracy when compared with MFCC and wavelet transforms alone, for both clean and noisy environments. The work presented in [41] used MFDWC for the Persian language. This research compared the results obtained by the MFDWC and the MFCC, for both clean and noisy speech signals. The results obtained confirmed that MFDWC performed better than MFCC, for both clean and noisy environments.

2.1.3 Wavelet packet transform: The WPTs are similar to DWT, with the only difference that both the approximation and detail coefficients are decomposed further [16].

The research presented in [13] compares a number of DFT and DWPT feature extraction methods for the SR task. One of the DFT methods considered in this study is the MFCC. The results obtained showed that the DWPT methods obtained higher recognition rates when compared with the DFT methods considered. Considering a DWPT-based method, a reduction in the word error rate of approximately 20% was achieved, when compared with the MFCC.

Another important comparison is that of WPT with DWT. When WPT was compared with DWT for the task of ASR, the performance obtained from the DWT outperformed that obtained from the WPT. This was shown in the work presented in [16], where comparison between the DWT and WPT for the Malayalam language is presented. The accuracies obtained for the WPT and DWT are 61 and 89%, respectively, showing a significant improvement in the recognition rate, when comparing DWT with WPT.

2.1.4 Linear predictive coding: The LPC method is a time domain approach, which tries to mimic the resonant structure of the human vocal tract when a sound is pronounced. LPC analysis is carried out by approximating each current sample, as a linear combination of P past samples, as defined by (7) [8, 10]

$$\hat{s}[n] = \sum_{k=1}^{P} a_k s(n-k) \tag{7}$$

This is obtained by first generating frames for the input speech signal, and then performing windowing of each frame in order to minimise the discontinuities present at the start and end of a frame. Finally, the autocorrelation between frames is evaluated, and the LPC analysis is performed on the autocorrelation coefficients obtained, by using Durbin's method [8, 33, 47].

LPC was first proposed in 1984 [48], but is still widely used nowadays [5, 33, 47, 49]. In the work presented in [33], the LPC are combined with the DWTs. After decomposing the input speech signal using DWT, each sub-band is further modelled using LPC. A normalisation parameterisation method, the CMN, is also used to make the designed system robust to noise signals. The proposed system is evaluated on isolated digits for the Marathi language, in presence of white Gaussian noise. The results obtained with this proposed feature extraction method, outperformed the results achieved with MFCC alone and MFCC along with CMN, by approximately 15%. Another work that also used LPC with DWT is presented in [5].

2.1.5 Linear predictive cepstral coefficients: The LPCC is an extension of the LPC technique [8]. After completing the LPC analysis, a cepstral analysis is executed, in order to obtain the corresponding cepstral coefficients. The cepstral coefficients are computed through a recursive procedure, as shown in (8) and (9) below [50].

$$\hat{v}[n] = \ln(G), \qquad \text{for } n = 0 \tag{8}$$

$$\hat{v}[n] = a_n + \sum_{k=1}^{n-1} \left(\frac{k}{n}\right) \hat{v}[k] a_{n-k}, \qquad \text{for } 1 \le n \le p \tag{9}$$

A recent research that studied the LPCCs for the task of ASR is presented in [51]. The proposed system studied the LPCC and MFCC, along with a modified self-organising map (SOM). The designed system is evaluated with 12 Indian words from five different speakers, and the results obtained showed that both LPCC and MFCC obtained similar results.

Another work that performed a comparison of the LPCC with MFCC is presented in [52]. This research analysed these two feature extraction techniques along with a simplified Bayes decision rule, for the speech recognition of Mandarin syllables. The results obtained showed that the LPCC achieved an accuracy which is 10% higher than that obtained by the MFCC. Additionally, the extraction of the LPCC features is 5.5 times faster than the MFCCs, resulting in lower computational time.

2.1.6 Perceptual linear prediction (PLP): The PLP is based on three main characteristics: spectral resolution of the critical band, equal loudness curve adjustment and application of intensity-loudness power law, in order to try and mimic the human auditory system. The PLP coefficients are obtained by first performing FFT on the windowed speech frame, and then apply the Bark-scale filtering, shown in (10), where B is the Bark-warped frequency. The Bark-scale filtering executes the first characteristic of the PLP analysis, since it models the critical band frequency selectivity inside the human cochlea [8, 13, 50].

$$\theta(B_i) = \sum_{B=-1.3}^{2.5} \left|X(B - B_i)\right|^2 \psi(B) \tag{10}$$

Afterwards, the Bark-scale filtering outputs are weighted according to the equal-loudness curve, and the resultant outputs are compressed by the intensity-loudness power

law. Finally, the PLP coefficients are computed by performing consecutively on the filtering outputs the inverse Fourier transform, the linear predictive analysis and the cepstral analysis [8, 13, 50].

The research presented in [13], performed the PLP features with two different window lengths. The TIMIT Corpus was utilised for the evaluation of this research, and the available phonemes were clustered into 38 classes. As for the classification stage of the ASR system, the HMMs were employed. The results obtained showed that for a window length of 25.625 ms, the PLP has approximately the same word and sentence error rates as the MFCC. However, when the window length was reduced to 16 ms, the recognition rates of the MFCC improved slightly, whereas those obtained by the PLP analysis remained the same. Hence, this resulted into the MFCC achieving a reduction in the word and sentence error rates, of approximately 1.1 and 2.3%, respectively, when compared with the PLP.

The PLP analysis was also employed for the recognition of Malay phonemes [53]. In this research, instead of utilising the PLP feature vectors, the PLP spectrum patterns were used. Hence, the recognition of phonemes was obtained through speech spectrum image classification. These spectrum images were inputted into an MLP network, for the recognition of 22 Malay phonemes, obtained from two male child speakers. With this approach, the accuracy reached was that of 76.1%.

Considering the implementation of PLP analysis in noisy environments, the work presented in [54], studied the PLP analysis along with a hybrid HMM–ANN system, for the task of phoneme recognition. The TIMIT Corpus was employed for evaluation, and the phonemes available were folded to a total of 39 classes. With this approach, the authors succeeded in achieving a recognition rate equal to 64.9%. However, when this system was evaluated with the handset TIMIT (HTIMIT) Corpus, which is a database of speech data collected over different telephone channels, the accuracy was degraded to 34.4%, owing to the distortions that are present in communication channels. In research [55], two different noise signals: white noise and street noise were considered for the task of word recognition of six languages: English, German, French, Italian, Spanish and Hungarian. The results obtained showed that both PLP and MFCC achieved approximately the same accuracies. Nevertheless, the PLP analysis performed slightly better than the MFCC, in clean, white and street noises, by approximately 0.2%. The authors state that this slight improvement of PLP with respect to MFCC could be attributed to the critical band analysis method. Apart from this, in research [50], it was proved that the PLP performs also better than the LPCC, when it comes to noisy environments.

2.1.7 RelAtive SpecTrA–perceptual linear prediction (RASTA–PLP): The RASTA–PLP analysis consists in the merging of the RASTA technique to the PLP method, in order to increase the robustness of the PLP features. The RASTA analysis method is based on the fact that the temporal properties of the surrounding environment are different from those of a speech signal. Hence, by band-pass filtering the energy present in each frequency sub-band, short-term noises are smoothed, and the effects of channel mismatch between the training and evaluation environments are reduced [8, 10].

The work presented in [54], apart from considering the PLP features, as explained in Section 2.1.6, the RASTA–PLP technique was also studied. From the results obtained, it can be concluded that for clean speech, the RASTA–PLP achieved a lower recognition rate, of 3.7%, when compared with the PLP method. However, when the HTIMIT was considered, the RASTA–PLP outperformed PLP, by obtaining an increase in the accuracy equal to 11.8%. Hence, this research confirms that when it comes to noisy environments, the addition of RASTA method to the PLP technique, results in feature vectors that are more robust.

Another research which demonstrates the robustness of the RASTA–PLP over the PLP technique is presented in [56]. In this work, two different experiments were studied. The first experiment considers these two feature extraction techniques, along with a CDHMM, for small vocabulary isolated telephone quality speech signals. With both training and test sets having the same channel conditions, RASTA–PLP performs only slightly better than the PLP. However, when the test data was corrupted, the RASTA–PLP outperforms PLP by 26.35%. To better confirm the results obtained above, the authors collected a number of spoken digits samples, over a telephone channel under realistic conditions. As expected, the RASTA–PLP obtained again a higher recognition rate when compared with the PLP features, which is approximately equal to 23.66% higher. For this task only, the LPC features were also considered. However, the LPC features achieved the lowest accuracies, with a reduction of 29.73 and 53.03%, when compared with the PLP and RASTA–PLP, respectively. As for the second experiment, the DARPA Corpus was utilised, in order to test with large vocabulary continuous high-quality speech. For this experiment, the CDHMMs were changed with a hybrid HMM–ANN system, and low-pass filtering was applied to the speech signals, in order to add further distortions. The results obtained showed that when the low-pass filtering was applied, the accuracy obtained from the PLP features decreased by 46.8%, whereas that achieved by the RASTA–PLP was reduced only by 0.6%.

The RASTA–PLP analysis was also considered with wavelet transforms, for the Kannada language [57]. Three different feature extraction techniques: LPC, MFCC and RASTA–PLP, were examined for the recognition of isolated Kannada digits. However, before employing these techniques, the speech signals were pre-processed through the use of wavelet transforms. For clean speech, the DWT was used, whereas for noisy speech the WPT was employed for pre-processing and also for noise removal. The results obtained confirmed, that by applying wavelet transforms to other feature extraction techniques, an improvement in the accuracies is obtained. For clean speech, the RASTA–PLP method alone achieved the lowest accuracy, equal to 49%, followed by the LPC, with 76%, and finally the MFCC, with the highest accuracy, equal to 81%. With the addition of the DWT, all three accuracies were increased, with MFCC, LPC and RASTA–PLP, achieving 94, 82 and 52%, respectively. Considering noisy speech, RASTA–PLP achieved the highest accuracy, equal to 73%, followed by the MFCC with 60% and finally the LPC, which achieved an accuracy of 53%. When WPT was considered, all accuracies were improved, but RASTA–PLP achieved the highest accuracy, which was equal to 83%.

Hence, it can be concluded that when it comes to clean speech signals, the RASTA–PLP method, may not be the best choice. Even when, both training and test environments are similar, the RASTA–PLP will only slightly improve the

doi: 10.1049/iet-spr.2012.0151

accuracies, when compared with the PLP features. However, for noisy environments, the RASTA–PLP outperformed the PLP, the LPC and the MFCC features. The robustness of the RASTA–PLP was also further improved, when combined with wavelet transforms.

2.1.8 Vector quantisation: The objective of VQ is the formation of clusters, each representing a specific class. During the training process, extracted feature vectors from each specific class are used to form a codebook, through the use of an iterative method. Thus, the resulting codebook is a collection of possible feature vector representations for each class. During the recognition process, the VQ algorithm will go through the whole codebook in order to find the corresponding vector, which best represents the input feature vector, according to a predefined distance measure. The class representative of the winning entry in the codebook will be then assigned as the recognised class representation for the input feature vector. The main disadvantage of the VQ method is the quantisation error, because of the codebook's discrete representation of speech signals [2, 42].

The VQ approach is also used in combination with other feature extraction methods, such as MFCC [58] and DWT [5, 42], in order to further improve the designed ASR system by taking advantage of the clustering property of the VQ approach.

2.1.9 Principal component analysis (PCA): PCA is carried out by finding a linear combination with which the original data can be represented. The PCA is mainly used as a dimensionality reduction technique at the front-end of an ASR system. However, the PCA can also be utilised for features de-correlation, by finding a set of orthogonal basis vectors, where the mappings of the original data to the different basis vectors are uncorrelated [8, 59, 60].

Various researches employed the PCA, in order to increase the robustness of the designed system under noise conditions [59–61]. In research [59], the authors state that the PCA analysis is required, when the recognition system is corrupted by noisy speech signals. This statement is confirmed through an evaluation made on four different noisy environments, employing Nevisa HMM-based Persian continuous speech recognition system. The results obtained showed that when the PCA was combined with the CMS to a parallel model combination, the robustness of the recognition system was increased. Another recent research, proposed a PCA-based method, with which further reduction in the error rates was obtained [60]. This PCA-based approach was also combined with the MVN method, in order to make the proposed recognition system more robust. This approach was evaluated with the Aurora-2 digit string corpus, and the results obtained showed that this approach achieved a reduction in the error rates of approximately 18 and 4%, with respect to the MFCC analysis, and when employing only the MVN method, respectively.

The PCA was also combined with the MFCC, in order to increase the robustness of the latter technique [61]. As stated in the section discussing MFCC, one of its drawbacks is its low robustness to noise signals. Hence, in this research, the MFCC algorithm is modified by computing the kernel PCA instead of the DCT. Thanks to the kernel PCA, the recognition rates obtained with noisy speech signals, were increased from 63.9 to 75.0%. However, when it comes to clean environments, the modified MFCC obtained similar results to the baseline MFCC.

2.1.10 Linear discriminant analysis (LDA): LDA is another dimensionality reduction technique, as the PCA. However, in contrast to PCA, the LDA is a supervised technique [8]. The concept behind LDA is the mapping of the input data to a lower-dimensional subspace, by finding a linear mapping that maximises the linear class separability [62]. The LDA is based on two assumptions: the first one is that all classes have a multivariate Gaussian distribution, and the second assumption states that these classes must share the same intra-class covariance matrix [63].

Various modifications were proposed to the baseline LDA technique [62, 64]. One popular modification is the heteroscedastic LDA (HLDA), in which the second assumption of the conventional LDA is ignored, and thus each class can have a different covariance matrix [63]. The HLDA is then used instead of the LDA, for feature-level combination [63, 64]. Another recent modification is proposed in [62], where this time, the first assumption of the baseline LDA is modified. In this research, a novel class distribution, based on phoneme segmentation is proposed. The results obtained showed, that comparable or slightly better results were obtained, when compared with the conventional LDA.

2.2 Classification

Numerous researches have been carried out in order to find that ideal classifier which recognises correctly speech segments under various conditions. Three renowned methods that were used at the classification stage of ASR systems are the HMM, the ANN and the SVMs. In the following section, these three methods will be discussed with respect to their implementation in the field of ASR.

2.2.1 Hidden Markov models: HMM is the most successful approach, and hence the most commonly used method for the classification stage of an ASR system [2, 10, 65–67]. The popularity of HMMs is mainly attributed to their ability in modelling the time distribution of speech signals. Apart from this, HMMs are based on a flexible model, which is simple to adapt according to the required architecture, and both the training procedure and the recognition process are easy to execute. The result is an efficient approach, which is highly practical to implement [2, 10, 68, 69].

In simple words, with HMMs the probability that a speech utterance was generated by the pronunciation of a particular phoneme or word can be found. Hence, the most probable representation for a speech utterance can be evaluated from a number of possibilities [2]. Consider a simple example, of a first-order three-state left-to-right HMM, as shown in Fig. 3. The left-to-right HMM is the type of model, which is commonly employed in ASR applications, since its configuration is able to model the temporal characteristics of speech signals. An HMM can be mainly represented by three parameters. First, there are the possible state transitions that can take place, represented by the flow of arrows between the given states. Each of these state transitions are depicted by a probability, a_{ij}, which is the probability of being in state S_j, given that the past state was

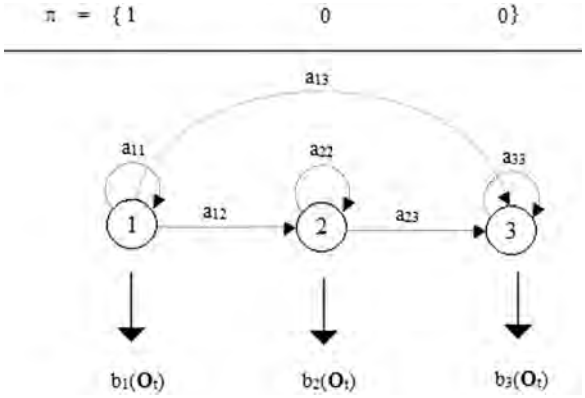

Fig. 3 *First-order three-state left-to-right HMM [68, 70]*

S_i, as shown in (11) [68, 70]

$$a_{ij} = P\left(q_t = S_j | q_{t-1} = S_i\right) \tag{11}$$

Second, there are the possible observations that can be seen at the output, each representing a possible sound that can be produced at each state. Since the production of speech signals differs, these observations can be also represented by a probabilistic function. This is normally represented by the probability variable $b_j(\boldsymbol{O}_t)$, which is the probability of the observation at time t, for state S_j. At last, the third parameter of an HMM is the initial state probability distribution, π. Hence, an HMM can be defined as [68, 70]

$$\lambda = (A, b, \pi) \quad \text{for } 1 \le i, j \le N \quad \text{and} \quad 1 \le k \le M \tag{12}$$

where $A = \{a_{ij}\}$, $B = \{b_j(\boldsymbol{O}_t)\}$, N is the number of states, and M is the number of observations. Consequently, the probability of an observation can be determined from [68, 70]

$$P_r(O|\pi, A, B) = \sum_q \pi_{q_t} \prod_{t=1}^{T} a_{q_{t-1}} b_{q_t}(O_t) \tag{13}$$

The groundwork of HMMs is based on three fundamentals, namely the evaluation of the probability for a sequence of utterances for a given HMM, the selection of the best sequence of model states, and finally the modification of the corresponding winning model parameters for better representation of the speech utterances presented [71]. For further theoretical details on HMMs, interested readers are referred to [68, 70, 71].

Some of the work done for continuous phoneme recognition will now be discussed. Particular consideration is given to the task of phoneme recognition since with HMMs, words are always based on the concatenation of phoneme units. Hence, adequate word recognition should be obtained if good phoneme recognition is achieved [23, 72].

One of the early papers, which proposed the use of HMMs for the task of phoneme recognition, considered discrete HMMs [72]. Discrete HMMs were designed along with three sets of codebooks, for the task of speaker-independent phoneme recognition. The codebooks consist of various VQ LPC components, which were used as emission probabilities of the discrete HMMs. A smoothing algorithm, with which adequate recognition can be obtained even with a small set of training data, is also presented. Two different phone architectures were considered: – a context independent model and a right-context-dependent model. The resultant phoneme recognition system was evaluated with the TIMIT database, where the phonemes were folded to a total of 39 classes according to the CMU/MIT standards. The highest results were obtained from the right-context-dependent model, with a percentage correct equal to 69.51%. With the context-independent model, a percentage correct of 58.77% was achieved. With the addition of a language model, bigram units were considered, and the percent correct increased to 73.80 and 64.07%, for the right-context-dependent and context-independent models, respectively. Additionally, a maximum accuracy of 66.08% was achieved from the right-context-dependent model, when considering also the insertion errors.

A popular approach is the use of phone posterior probabilities. Recent studies that work with phone posteriors are presented in [26, 73]. The standard approach is based on the use of MLP to evaluate the phone posteriors [74]. Spectral feature frames are inputted to an MLP, and each output of the MLP corresponds to a phoneme. The MLP is then trained to find a mapping between the spectral feature frames presented at the input, and the phoneme targets at the output. Afterwards, a logarithmic function and a Karhunen–Loeve transform (KLT) are performed on the MLP phone posterior probabilities, to form the feature vectors, which will be presented to an HMM, for training or classification. In [73], two approaches for enhancing phone posteriors were presented. The first approach initially estimates the phone posteriors using the standard MLP approach, and then uses these as emission probabilities in the HMMs forward and backward algorithm. This results into enhanced phone posteriors, which take into consideration the phonetic and lexical knowledge. In the second approach, another MLP post-processes the phone posterior probabilities obtained from the first MLP. The resultant phone posteriors from the second MLP are the new enhanced phone posterior probabilities. In this manner, the inter- and intra-dependencies between the phone posteriors are also considered. Both approaches were evaluated on small and large vocabulary databases. With this approach, a reduction in the error rate was obtained, for frame, phoneme and word recognition rates. Apart from this, the resultant increase in computational load due to the enhancement process is negligible. Another research proposes a two-stage estimation of posteriors [26]. The first stage of the designed system is based on a hybrid HMM–MLP architecture, whereas the second stage is based on an MLP with one hidden layer. For the hybrid HMM–MLP architecture, both context-independent and context-dependent HMMs were considered. Comparing the results obtained from these two researches [26, 73], both systems were evaluated with the TIMIT database, and clustered the phonemes to a total of 39 classes. The enhanced phone posteriors approach proposed in [73], achieved a phone error rate of 28.5%. However, a better result was obtained with the two-stage estimation of posteriors proposed in [26], where a phone error rate of 22.42% was achieved.

A procedure based on HMMs and wavelet transforms was also proposed in [75], in order to improve wavelet-based algorithms by making use of the HMMs. This method is called the hidden Markov tree (HMT) model. Wavelet transform algorithms have already proved their ability in

doi: 10.1049/iet-spr.2012.0151

providing excellent results in speech recognition applications. However, regular wavelet transform algorithms treat each wavelet coefficient independently. If the dependencies between wavelet coefficients are also considered, their performance might improve. With HMT model, simple Markov structures are used to model the dependencies between the wavelet coefficients. These Markov structures are applied between the states of the wavelet coefficients and not directly to the wavelet coefficients. The end result is a binary tree structure, with the wavelet states connected vertically across the scale. The HMT model was evaluated on a simple signal classification problem. The results obtained showed that a further reduction in the error rate was obtained when comparing the HMT model to a wavelet-based algorithm. Apart from this, different noise signals were also considered, and the results obtained showed that better de-noising was achieved with the HMT model [75]. Hence, the HMT model is also suitable for robust ASR. In fact, an enhanced method for the utilisation of the HMT model for de-noising applications is presented in [76]. The proposed method is made up of two cascaded stages: the first one being the HMT model for the de-noising process and at the second stage a weighted filter bank analysis is performed. The proposed feature extraction method was evaluated on noisy speech signals with SNR from 25 to 0 dB. Comparing the proposed method with the HMT model presented in [75], the former approach achieved a higher recognition rate, up to an SNR of 20 dB. For example, with an SNR of 10 dB, an increase in word and sentence recognition rates of approximately 4 and 7% was achieved, respectively. However, for SNR of 25 dB the same recognition rate as the HMT model was obtained. Hence, for high SNRs it might be more suitable to opt for the HMT model presented in [75], which implements a simpler feature extraction method. Another research considers the implementation of HMT models as emission probabilities of a HMM [40]. An important drawback of HMT models in speech recognition tasks is the inability of handling sequences of variable length. However, this is not an issue when merging HMT with HMMs. Additionally, the performance of HMMs is also improved, since with HMT models as emission probabilities of HMMs, the assumption of stationarity is removed. Hence, in this research [40] an expectation–maximisation (EM) algorithm is proposed, which uses the output observations from the HMT model, at each state of the HMM. In this manner, the HMM will take care of the long temporal information, whereas the local dynamics are captured in the wavelet domain by the HMT models. The designed system was evaluated on five different phonemes from the TIMIT database. The accuracies obtained showed that the proposed system achieved a higher recognition rate, when compared with a Gaussian multi mixture (GMM) model, an HMT-model, and an HMM with GMM as emission probabilities. The HMM–HMT model achieved an accuracy rate equal to 42.38%, exceeding the HMM–GMM model, the GMM model and the HMT model, by 11.54, 9.76 and 4.29%, respectively.

Recently, the major approach used in ASR system is the CDHMM [23, 45, 77]. CDHMM is based on an efficient maximum likelihood (ML) algorithm for the training and recognition of the HMMs. With CDHMM, one is able to capture the variations between and within phonemic units [23].

The work presented in [23] tackles the concept of large vocabulary in continuous SR (LVCSR) with CDHMM. As research advances in the field of ASR, the concept of LVCSR is becoming more prominent. The researchers of this work state that it is more feasible to tackle the problem of LVCSR at phonemes level rather than at words level, since the number of phonemes is less than the number of possible words in a large vocabulary. Hence, in this study a speaker-independent phoneme recognition system for continuous speech environments is designed based on CDHMM. The aim of this research was to find the optimal model architecture with which a robust phoneme recognition system is achieved, but at the same time keeping in mind the limited amount of training data available. The approach considered is based on the idea of increasing the number of Gaussian mixture components per state according to the corresponding number of frames available for training. In this manner, the resolution of the CDHMM models is increased or decreased until the required performance level is reached. Two different initialisation methods were employed for the model's output probabilities. The first initialisation method is based on the flat start procedure, whereas the second initialisation approach is based on the Viterbi algorithm. The HMM toolkit (HTK) [78] was used throughout this research for the design of the CDHMM, and at the feature extraction stage the MFCC was considered. The TIMIT database was used to evaluate the designed phoneme recognition system. The results obtained showed that the second initialisation approach performed best, with a percentage correct and percentage accuracy equal to 60.68 and 54.01%, respectively. As the number of Gaussian mixtures was increased, a noticeable improvement in the accuracies was obtained. With 64 Gaussian mixtures per state, an accuracy of 67.79% was achieved. However, this increase in accuracy comes at a cost, since the complexity of the system increases as the number of Gaussian mixtures is increased.

Another approach that was proposed to improve further the CDHMM, was the training procedure based on the concept of large margin classifiers, which is used in machine learning. This method already proved its ability in reducing the error rates, when compared with the conventional ML estimator [77, 79, 80]. The origin of this concept was proposed in [79]. This work was further improved in [80], by adding a margin-based cost function that penalises a data point, which was incorrectly classified according to its Hamming distance from the desired transcription. Another work proposed the idea of optimising not only the mean parameters of the GMMs during the training phase, but also the variance parameters [77]. With the consideration of the variance parameters, a further reduction in the recognition error rates was obtained. The research presented in [81] proposes a different approach to large margin CDHMMs based on a Bayesian learning method. Apart from this, an improvement to the ASR system to deal with different testing environments is also shown. The designed system was tested for phoneme recognition with the TIMIT database and the results obtained showed a slight reduction of approximately 1% in the phoneme error rates when compared with large margin CDHMMs proposed in [79, 80]. However, comparison with the work done in [77] was not given.

Although huge improvements and significant recognition rates were obtained, HMMs are still far from achieving an optimal ASR system by themselves [10, 69]. One of the major limitations with HMMs is the assumption that the probability of being in a particular state is only dependent

on its preceding state, ignoring any long-term dependencies. This assumption is what makes HMMs simple to implement, nevertheless, it makes HMMs inaccurate. Additionally, the emission probabilities chosen for the HMMs states are arbitrarily chosen, and might not even represent the output probabilities of the corresponding state properly. In general, these emission probabilities are represented by GMMs, where the number of mixtures chosen might either limit the system, or increase its complexity unnecessarily [82]. Hence, although considerable accuracies were obtained from ASR systems based on HMMs, one still has to find either a way of improving the present HMMs systems or establish a completely new approach for ASR applications.

2.2.2 Artificial neural networks: ANNs are the second most widely used method at the classification stage of an ASR system. These were used either independently or as a combination with HMMs, the latter being the one which is extensively used, in order to combine the advantages of both ANNs and HMMs [2, 10, 45].

ANNs are excellent classifiers and are highly adequate for pattern recognition applications. They are desirable to use, because of their ability in organising and learning according to the datasets inputted during the training phase. Additionally, ANNs are capable to adapt when unknown data are presented, thus being able to classify new data effectively [16, 45]. However, ANNs are based on Empirical Risk Minimisation (ERM), which makes ANNs prone to over training and local minima problems [45]. Another drawback is their inability in representing the time variability present in speech signals. This drawback is generally solved with the consideration of a hybrid model, where ANNs and HMMs are merged together [2].

Five ANN architectures that are widely used nowadays are briefly discussed, focusing mainly on recent research. These are the MLP, the self-organising maps (SOMs), the radial basis function (RBF), the recurrent neural network (RNN) and the fuzzy neural network (FNN).

2.2.2.1 Multilayer perceptrons: The MLP is the most successful and hence the most popular ANN architecture in the field of ASR [2, 26]. Basically, the MLP is a feed-forward network consisting of at least three layers:– the input layer, the hidden layer and the output layer. A simple example of an MLP is shown in Fig. 4.

The learning algorithm employed, during the training process, is generally based on the conventional backpropagation approach and the concept of lateral inhibition. Afterwards, during the recognition phase, the resultant output is determined according to the representation corresponding to the output neuron, which results in the highest activation. A major drawback of MLP is the inability to handle the dynamicity of speech signals. As a result, inputs presented to MLP need to be of fixed length. Apart from this, MLPs are only able to deal with small vocabularies, making them more appropriate to phoneme recognition rather than word recognition [1, 2, 83].

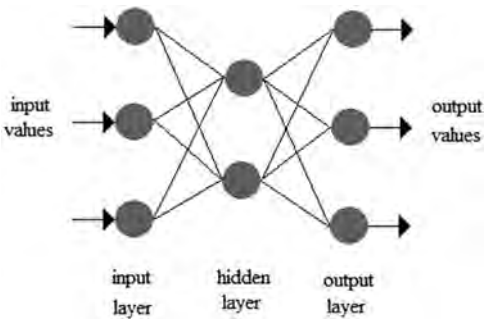

Fig. 4 *Multilayer perceptron [83]*

Looking at some work which considered the use of MLP, in [84], the MLP were used to recognise Urdu digits from a mono-speaker database, under noise-free conditions. As feature parameter extractors, the FFT and MFCC were employed. With these two feature parameter extractors, and the MLP for classification, an accuracy of 94% was reached. The work in [16] proposed also the use of MLP with either the WPT or the DWT as feature extractors, for the Malayalam language. This research mainly concentrates on the comparison of the WPT and DWT implementations. However, the results obtained showed that the MLP can be successfully embedded with wavelet transform approaches. The MLP were also utilised for the recognition of Persian digits [85]. In the proposed method, the speech signal is first de-noised with MFCC, and afterwards feature vectors were extracted with the use of DWT. The feature vectors obtained were then inputted to the MLP for classification. For Persian spoken digits from a male speaker, an accuracy of 98% was achieved.

As already mentioned previously, the MLP are widely used to evaluate the posterior probabilities, in order to further improve the accuracies obtained from HMMs. Considering the baseline system presented in [26], comparing HMMs and a hybrid HMM–MLP architecture, a lower phoneme error rate equal to 6.27%, was obtained with the latter approach. Apart from improving HMMs-based systems, this hybrid approach resulted also in the utmost successful results obtained from MLP-based architectures [26].

Recently, there was also the introduction of a new concept based on MLP, the sparse MLP (SMLP) [83, 86]. The SMLP are based on the same layout as MLP, with the only difference that the outputs of one of the hidden layers are sparse. A phoneme recognition system, based on SMLP is introduced in [83]. The designed phoneme recognition system is based on a hybrid HMM–SMLP approach, where the posterior probabilities are dependent on the sparse hidden features. The feature parameter extractor employed is based on PLP analysis. The designed system was evaluated on the TIMIT database, where the total number of phoneme classes were reduced to 49 for training, and then further reduced to 39 classes for classification according to the CMU/MIT standards. A phone error rate of 21.2% was obtained, with a 6.2% improvement with respect to MLP-based systems. This system was further improved in [86], by considering also the frequency-domain linear prediction (FDLP) temporal features and the modified linear discriminant analysis (MLDA) spectro-temporal features, as feature parameter extractors, with the previous PLP cepstral coefficients. With the new feature vectors, the phoneme error rate was further reduced to 19.6%.

2.2.2.2 Self-organising maps: The SOM was introduced by Kohonen in 1982. The basic idea behind SOMs is the clustering of data in such a way to produce a topographic map from a high-dimensional input space to a lower-dimensional feature space. In this manner, all of the input data that are initially randomly localised in the input feature space, and are then organised into clusters, each representing a distinct feature of the input data. Hence, the

doi: 10.1049/iet-spr.2012.0151

SOMs are capable of distinguishing between the main characteristics of the input data presented to them [22, 51, 87].

The SOMs achieve this through an unsupervised learning process, which therefore takes place without having available, any target output to compare with. Hence, a substantial number of samples are required, for the SOM network to be trained adequately. Basically, the SOMs learning algorithm consists of three steps. The first step is the competitive learning process, where the similarity between the neurons in the output layer, and the input pattern presented to the network, is computed according to a pre-determined function, which is normally the Euclidean distance. In the second step the lateral inhibition approach is employed and finally there is the adaptation of the synaptic weights, as shown in (14), where the synaptic weight vector $\boldsymbol{w}_j(n)$ of neuron j at time n, is updated to $\boldsymbol{w}_j(n+1)$ at time $n+1$, and $\alpha(n)$ and $h_{j,i(x)}$ are the learning rate parameter and neighbourhood function, respectively [87, 88].

$$\boldsymbol{w}_j(n+1) = \boldsymbol{w}_j(n) + \alpha(n)h_{j,i(x)}(n)\Big(x - \boldsymbol{w}_j(n)\Big) \tag{14}$$

The work presented in [89] proposes the implementation of an SOM with the use of wavelet transforms for the task of vowel recognition. The designed system, referred to as the wavelet SOM (WSOM), uses a SOM to model the input data and adapt wavelets according to the resultant SOM mapping. With WSOM an accuracy of 55% was achieved for the task of vowel recognition.

Another research proposes the use of SOMs with HMMs, in order to make the designed system adequate for real-time applications [22]. Hence, after forming the required HMM models, these were clustered with an SOM. Therefore during the classification process, first the ideal SOM cluster was chosen, and then a HMM from the corresponding cluster is adopted as the final model. The proposed system was evaluated on a speaker-dependent spoken digits recognition task.

SOMs were also used to change variable length feature vectors to fixed length, as presented in [90]. The designed system uses the MLP for classification, and as mentioned earlier, MLP are not capable to handle variable length feature vectors. Hence, SOMs are employed at the pre-processing stage, in order to adjust the length of the feature vectors to a predefined fixed length, before being fed to the MLP for recognition.

A recent research that adopts the SOMs for the task of ASR is presented in [51]. The basic SOM is modified into a supervised SOM, consisting of an input layer, a competitive layer and an output layer. Four different features, the LPCC, the MFCC, the pitch, and the intensity, were considered at the feature extraction stage of the ASR model. The designed system is evaluated with 12 Indian words from five different speakers. The accuracies obtained from the five different speakers were analysed independently. Considering the mean-SOM performance, and the median-SOM performance with respect to the accuracies obtained from all the speakers, the vocal intensity feature obtained the highest recognition rate. The accuracies obtained from the mean-SOM and the median-SOM for the intensity features are 98.17 and 98.54%, respectively. As regards to the LPCC, the MFCC and the pitch features, the three of them achieved approximately an accuracy of 89%.

As can be noted from the researches discussed above with respect to the implementation of SOMs for ASR applications, there is still ample of work to be carried out. Until now, SOMs have been evaluated only with small vocabularies and a small amount of speakers.

2.2.2.3 Radial basis functions: An RBF architecture, basically consists of three layers: the input layer, the hidden layer and the output layer. The key element of an RBF model lies in the procedure performed in the hidden layer, where a Gaussian function is utilised. The concept behind an RBF network is the generation of clusters based on the patterns present in the input data. The relationship of an unseen input to the clusters formed is then computed by means of the Gaussian function from the centres of these clusters. Hence, the output of an RBF network, which consists of H nodes in the hidden layer, for an input x, is defined by

$$y = \sum_{h=1}^{H-1} w_h \phi_h(x) \tag{15}$$

where w_h are the linear weights, and ϕ_h is the Gaussian function, which can be further defined as

$$\phi_h = \mathrm{e}^{\left(\|x-c_h\|/2\sigma_h^2\right)} \tag{16}$$

where c_h and σ_h are the centre and width of the Gaussian function, respectively. In recent years, RBFs are becoming popular and are being widely used in different applications. As a matter of fact, RBFs are also proving to be a good alternative to the present popular MLPs [91].

A recent research which compares the implementation of RBFs to MLPs, for the task of isolated word recognition is presented in [91]. The proposed ASR system employs the LPCC for feature extraction, and analysis an RBF network and an MLP network for the classification stage. The designed system was evaluated on six English words spoken by six speakers. The accuracies obtained for the MLP and RBF networks are 96 and 98.69%, respectively. Additionally, when compared to MLP, the training and testing speeds of the RBF architecture are faster.

The work presented in [92] proposes an implementation of the RBF with HMMs, for the task of word recognition in a continuous speech environment. The proposed ASR system first extracts features from the inputted speech signal through the use of cepstrum analysis, and these features are then fed to the hybrid HMM–RBF model. With this hybrid approach, an HMM is constructed for each word in the database, and a target value is associated with each HMM. Afterwards, for each of these target values, the optimum number of neurons in the hidden layer of the RBF network had to be found. For the classification of ten different words, the highest accuracy was obtained with eight neurons in the hidden layer, achieving an accuracy of 80%.

An implementation of the integration of RBFs with a wavelet transform was also proposed for noise robust ASR [93]. In this research, the standard RBF network is modified such that instead of using the common RBF activation function, a wavelet-based function is used. In this manner, the modified RBF network will benefit from the characteristics of the wavelet transforms, which make them highly robust to noise signals. This new approach was evaluated for the task of word recognition using

different amount of words from 16 speakers, in different SNR environments. The results obtained from the designed wavelet-RBF network, are superior to the ones obtained from a standard RBF architecture, for all the different SNRs considered. However, as the amount of words increased, the accuracy of the wavelet-RBF network decreased almost reaching the ones obtained with the RBF network. Hence, for large vocabularies it would be better to opt for the standard RBF network implementation, since at a cost of increased complexity with the wavelet-RBF network, negligible improvement in the accuracies was obtained.

2.2.2.4 Recurrent neural network: An RNN basically consists of three layers: the input layer, the hidden layer and the output layer. The concept behind RNN is the employment of feedback connections, either at the hidden or output layers. The output from the respective nodes, are multiplied by the corresponding weight, and fed back to the node itself. As a result, the state of a node is dependent not only on the present input, but also on the past state of the node [94, 95].

The RNN was also able to achieve better results than the MLP. Nonetheless, the training algorithm of the former approach is more complex, and also sensitive to any changes [94]. The work presented in [95] alters the MLP, which is a feed-forward network, into an RNN, by adding feedback connections either at the hidden or output layers. In this manner, this new MLP structure is able to handle the time variation of speech signals. In this research, two experiments were studied. The first method consists in classifying all the phonemes with only one network, whereas in the second experiment, the phonemes are grouped into six categories, based on the phonemes' types, and a separate RNN–MLP network is trained for each category. The proposed system was evaluated with 33 phonemes from the Japanese language, spoken by a male speaker. In both experiments, the RNN–MLP achieved better results than the conventional MLP. Apart from this, the RNN–MLP structure that has feedback connections at the output layer performed better than the configuration that has feedback connections at the hidden layer. In the first experiment, the RNN–MLP obtained an increase in the accuracy of 16.4%, when compared with the MLP. However, in the second experiment, all networks were able to achieve higher recognition rates. Nonetheless, the RNN–MLP was still able to surpass the MLP, with an accuracy of 10.6%. Another research that gives a comparison of the RNN and MLP networks is presented in [96]. In this research, the baseline RNN was improved by applying new training principles, based on deep learning, and by applying also second-order optimisation techniques. The Aurora2 corpus was employed for evaluation, and the results obtained confirmed that the RNN was able to achieve better results than the MLP. For the recognition of phonemes, the RNN outperformed the MLP, by an accuracy of approximately 7%. The proposed RNN network was also analysed in different SNR environments, and again the RNN was able to achieve lower error rates when compared to the MLP.

The RNN were also employed along with the probabilistic neural networks (PNN), for the task of speaker-independent phoneme recognition of Indian English speech [97]. The designed system employs first the PNN, in order to recognise to which category the inputted phoneme belongs to, and then the RNN is utilised to recognise exactly which phoneme it is. With this method, the authors were able to achieve an accuracy of 98%.

2.2.2.5 Fuzzy neural network: The FNN is based on the merging of fuzzy systems with neural networks. Thanks to fuzzy systems, a membership function is employed, in which an element is mapped to a proper degree of membership. This is optimal when it comes to speech recognition applications, since sounds in speech signals do not have clear boundaries [45]. Apart from this, an ANN is highly dependent on the amount of training data available. If this amount is not enough to train the network adequately, this may result in a poor-quality classifier. However, with the utilisation of FNN, the network will be able to converge during the learning process, resulting in better performance [98].

As with other neural networks, FNN can be employed either independently or as a hybrid, along with another technique. An example of a hybrid system, which includes an FNN, is presented in [45]. The proposed hybrid system consists of a wavelet transform, a CDHMM, and finally an FNN. The designed system was evaluated with a word list of 50 computer commands. From the results obtained, it can be concluded that when compared to a CDHMM system, the proposed hybrid architecture is more robust when it comes to noisy environments. As a matter of fact, the proposed approach was able to achieve an improvement in accuracy of 15.2%. However, in clean speech environments, the CDHMM performed better, with a positive difference of 7.6%, when compared with the proposed hybrid system.

A FNN-based system, which is widely employed in ASR systems, is the adaptive neuro fuzzy inference system (ANFIS) [98, 99]. This system applies a number of fuzzy inference techniques for data classification. The work presented in [98], considers the implementation of the ANFIS, for recognition of isolated Persian words. First, an SOM and a linear vector quantisation (LVQ) network are used for clustering of the input data, and then the ANFIS is employed for classification. The results obtained showed that classification with ANFIS, achieved better results, when compared with the conventional FNN. Another research which applies the ANFIS for speech recognition is presented in [99]. This time, the ANFIS was used for classification of speaker-independent isolated Malay digit speech signals, and a recognition rate equal to 85.24% was achieved. Although the above researches stated that good results were obtained with the ANFIS, none of these researches provided any comparison of the proposed systems with other ANNs.

2.2.3 Support vector machines: Recently, SVMs are also being adopted in ASR architectures, either independently, or as a hybrid architecture with HMMs [100–102]. The concept behind SVMs is the construction of a hyperplane, as the decision surface, such that the margin of separation between different classes is maximised. Under this condition, the resultant decision surface is defined as the optimal hyperplane. The decision surface can be defined as shown in (17), where $\boldsymbol{w}$ is the weight vector, b is the bias value and $\phi(\boldsymbol{x}_i)$ is the kernel function. Different kernel functions are used to map the input feature space to a higher-dimensional feature space, where the different classes are assumed to be linearly separable. The most popular kernel functions are the linear function, the polynomial function, the RBF and the two-layer perceptrons

doi: 10.1049/iet-spr.2012.0151

[103].

$$f(\boldsymbol{x}_i) = \boldsymbol{w}^{\mathrm{T}} \times \phi(\boldsymbol{x}_i) + b \tag{17}$$

The choice of the optimal hyperplane is highly dependent on a small subset of training data, referred to as support vectors. These support vectors are those data points that lie closest to the decision surface. For non-separable data, the construction of a decision surface without any classification errors is not possible. Nonetheless, an optimal hyperplane that minimises the probability of classification errors can be found [102–104]. For further theoretical details, interested readers are referred to [103, 104].

The two major problems with SVMs for ASR applications are the inability of handling variable inputs, and the high computational cost in classifying more than two classes at once. Over the past decades, a lot of research has been carried out in order to come out with the ideal solutions. In the following, various researches that tried to tackle the above two problems are discussed.

Numerous researchers came out with different approaches on tackling the problem of SVMs in classifying more than two classes [105–108]. However, the most popular approaches are those based on the reduction of a multiclass problem into a set of binary classes SVMs. The three multiclass SVMs methods that are extensively employed are namely the one-against-all, the one-against-one and the directed acyclic graph SVM (DAGSVM) [101, 102, 108–110].

With the one-against-all method, the multiclass problem is divided into a number of binary SVM classifiers, equal to the number of classes, which have to be distinguished. Each binary classifier constructs a hyperplane between its corresponding class and all the other classes. On the other hand, the one-against-one method, constructs a binary SVM classifier for each possible pair of classes, thus separating each class from each other. For both the one-against-all and one-against-one methods, a majority voting scheme is generally employed, in order to decide the output class for a given input [105, 108]. Comparing these two approaches, although the one-against-one method requires a higher number of binary SVMs classifiers than the one-against-all method, the size of the training dataset needed for each binary SVM for the one-against-one method is lower. Apart from this, in a one-against-one approach, if the two classes corresponding to a classifier are rarely required to be distinguished from each other, the corresponding binary SVM classifier can be ignored. All of this results in the one-against-one method having a lower computational cost [101, 103].

A drawback with multiclass SVMs being based on binary SVMs classifiers, is the unclassifiable regions problem. Unclassifiable regions are those regions in the feature space, undecided to which class they belong to. For the one-against-all method, this problem is solved through either the use of continuous decision functions or the implementation of fuzzy SVMs. Both methods are comparable with each other as regards to system's accuracies. However, the former approach is simpler to implement. As regards to the one-against-one method, the problem of unclassifiable regions is solved through the utilisation of the DAGSVM approach [111, 112]. In fact, the DAGSVM method is an improvement on the one-against-one method. The only difference between these two approaches is in the classification phase. Instead of a majority voting scheme, the DAGSVM utilises a rooted binary directed acyclic graph, where each node is a binary SVM. The resultant classification time is also reduced, when compared with the one-against-one method [107, 111].

Looking now at the inability of SVMs in handling variable input vectors, various approaches were proposed as a solution [102, 113]. Two widely used methods are the use of feature extraction techniques [102, 110] and HMMs [103, 113, 114].

The first approach is based on the employment of a feature extraction technique, in order to obtain a fixed length feature vector representing the speech signal, and present these feature vectors to the SVMs. An example of this approach, based on phoneme classification, is presented in [110]. The MFCC were used at the feature extraction stage, and the one-against-one method with majority voting was employed for classification. The designed system was evaluated on the TIMIT database, and compared with an HMM. With an HMM an accuracy of 73.7% was obtained, whereas with the designed SVM-based system an accuracy of 77.6% was achieved, resulting in an increase of approximately 4%. Hence, the SVM was even able to achieve a better accuracy than the HMMs.

Next approach is based on the employment of HMMs. When implementing a hybrid HMM–SVM system, two important issues have to be taken care of: the segmental modelling and the posterior estimation. Generally, with segmental modelling, a phoneme is divided into three segments and the ratio 3–4–3 is assumed. As regards to the posterior estimation, the Platt's posterior probabilities are usually employed, in order to change the output given by an SVM, which is usually a distance measure of the separation between the data point to be classified and the decision function, to a probabilistic value [103, 108, 114]. The work proposed in [101] utilises the one-against-one method, but instead of the majority voting scheme, the Platt's posterior probabilities are employed. These probabilities are then used in an HMM as the emission probabilities, instead of GMMs. After evaluating the designed ASR model on the DARPA Resource Management (RM1) corpus, the hybrid HMM–SVM obtained a reduction in the word error rate of up to 26%, when compared with the HMM–GMM baseline model.

As can be noticed, in recent years, researchers are also proving that the SVMs are also able to obtain either comparable or even better results than the well renowned HMMs. The work presented in [100], discusses the implementation of a structured SVM system, which is an extension of the baseline SVM system, and shows that better results can be obtained, when compared with HMMs. The designed system was evaluated with the TIMIT Corpus, where the total number of phoneme classes was reduced to 39 labels, according to the CMU/MIT standards. Different features were considered, and the structured SVM was able to achieve a slight improvement in the accuracy of 1.33%, with respect to the HMMs. Another recent research, in which the SVMs were compared with the HMMs, is presented in [115]. In this research, the SVMs were constructed based on the error-correcting output codes (ECOC), in which the division of multiclass methods into binary classifiers, is expressed in terms of matrix codes. Four different ECOC–SVM methods were analysed, all of which employed the RBF kernel. For evaluation, different vocabulary sizes of isolated words for the Korean language were utilised. In addition, apart from considering clean

speech, white Gaussian noise was also added to the testing samples, in order to evaluate the system at different SNR values. The results obtained showed that all the ECOC–SVM methods outperformed the HMMs, with the one-against-one method performing best, achieving approximately an improvement in the accuracy of 2–6%, depending on the vocabulary size and SNR value.

SVMs were also proved to perform better than ANNs. The superiority of SVMs mainly results from the fact that SVMs are based on structural risk minimisation (SRM). Owing to this, SVMs are not prone to over training and local minima problems, as is the case with ANNs [103]. The work presented in [116] presents a comparison of an SVM system, employing the one-against-all method with the RBF kernel, and an MLP network, for the recognition of 12 Thai vowels, spoken in isolation. The accuracies obtained were equal to 82.72 and 87.08%, for the MLP and SVM systems, respectively. In addition, the processing time for the training phase of the SVM system was shorter than that of the MLP. Another research that considered the recognition of isolated vowels is presented in [117]. This time 11 vowels of the British English language were considered. For the SVM system, the one-against-one and one-against-all methods were considered, and each of them was evaluated with both the linear and RBF kernels. As for the ANN, a three-layered ANN with different number of nodes in the hidden layer was considered. From the results obtained, it can be concluded that the ANN was able to perform better than the SVM, when the linear kernel was considered. The ANN obtained an improvement in the recognition rates of 5.8 and 6.1%, when compared to the one-against-one and one-against-all methods, respectively. However, when the SVM architectures were considered with the RBF kernel, these outperformed the ANN. The one-against-one and one-against-all methods were able to achieve an increase in the accuracy of 15.6 and 18.2%, respectively. A comparison of the SVM and ANN was also evaluated for the Hungarian numbers, with 28 phoneme classes [118]. Different feature extraction techniques were analysed, with both the SVM and ANN systems. The results obtained were approximately the same, for both the SVM and ANN systems. Nonetheless, the SVM system was able to achieve the highest recognition rate, with the Kernel-LDA feature extraction technique, which was approximately 2% higher than that achieved by the ANN. One more research that obtained comparable results between an SVM- and ANN-based systems, is presented in [119]. In this research, the SVM and an MLP network, were considered as hybrid architectures, along with the HMMs. Both systems were evaluated with the SpeechDat Spanish database, for large vocabulary continuous speech recognition.

Thanks to their generalisation capability, SVMs have already shown their ability in classification of speech signals [115, 120]. In fact, over the past years, numerous researches proved the superiority of SVMs over ANNs, but most importantly, SVMs were also able to obtain either comparable or even better results than the well-renowned HMMs.

2.3 Language model

Knowledge of the language being spoken is necessary for a spoken language system, in order to produce meaningful representations of the input speech signal [19]. With such knowledge, comprehensible human-like speech recognition can be obtained.

This knowledge can be divided into a number of levels. Starting from the lower level up to the higher level, the levels of linguistic analysis are as follows [19]:

1. Phonology: This level incorporates the knowledge about the linguistic sounds, within and across words. The variations in pronunciation, when words are from a continuous speech environment, are also taken care of at this level.
2. Morphology: Handling of the meaning of the components making up a word is dealt with at this level.
3. Lexical: This level focuses on the interpretation of the meaning of the words individually.
4. Syntactic: At this level, the words are analysed in the context of a sentence to determine the grammatical structure of the sentence. Hence, this level provides the knowledge with respect to the structural relationships between the words.
5. Semantic: All the possible meanings of a sentence are determined within this level.
6. Discourse: At this level, text longer than a sentence is considered. Hence, this level considers the meaning of the whole text.
7. Pragmatic: Within this level, world knowledge and understanding of the intentions, plans and goals of the speaker are required. Hence, at this level one also considers the subject on which the speaker is speaking about, when deciding on the interpretation of a particular word, if this word has several possible meanings.

All of the above levels can be implemented in an ASR system through the use of natural language processing (NLP). NLP consists of various computational techniques, which consider linguistic analysis to obtain human-like language processing. It has been shown that humans normally utilise all of the above levels. However, an NLP system may utilise one or more of the above levels, but not necessarily all of them. This can be deduced from the different NLP applications available. Nevertheless, the more levels of linguistic analysis an NLP system employs, the more the system will be capable of producing proper speech recognition. The amount of levels required are determined according to where the NLP is going to be applied. However, the lower levels of linguistic analysis are those which are widely studied and implemented [10, 19].

Different approaches on how to integrate an NLP to an ASR can be sorted into the following three categories [121]:

1. First the ASR model converts the speech signal into a sequence of phonemes or words, and afterwards the NLP attempts to understand the given words.
2. The ASR model outputs more than one possible representation of the speech signal. These are then analysed with an NLP and the best one is chosen.
3. The ASR model and the NLP are combined, such that the ASR model can make use of the information and constraints provided by the NLP.

A popular application is that of a dictionary, which consists of a list of all possible words that one might encounter in a particular application, along with their corresponding pronunciations. Current spoken language systems have limited vocabularies, since this is dependent on the available power and memory space of the central processing unit being used. As a result, one might encounter

doi: 10.1049/iet-spr.2012.0151

Table 1 Advantages and disadvantages of feature extraction techniques

Feature extraction technique	Advantages	Disadvantages
MFCC	• provides good discrimination [8] • low correlation between coefficients [8] • not based on linear characteristics; hence, similar to the human auditory perception system [8, 10] • important phonetic characteristics can be captured [8]	• low robustness to noise [8, 32] • in a continuous speech environment, a frame may not contain information of only one phoneme, but of two consecutive phonemes [27, 32] • limited representation of speech signals since only the power spectrum is considered, ignoring the phase spectrum of speech signals [8, 24]
DWT	• considers also temporal information present in speech signals, apart from the frequency information [27] • able to perform efficient time and frequency localisations [8, 39, 44] • successfully used for de-noising tasks [8, 45] • capable of compressing a signal without major degradation [8]	• not flexible since the same basic wavelets have to be used for all speech signals [8]
WPT	• same as DWT, but WPT shows also further detail present in the high frequency bands [16]	• not flexible since the same basic wavelets have to be used for all speech signals [8]
LPC	• spectral envelope is represented with low dimension feature vectors [8, 91] • good source-to-vocal tract separation is obtained [8] • LPC method is simple to implement and mathematically precise [8]	• linear scales are not adequate for the representation of speech production or perception [10] • Feature components are highly correlated [8] • cannot include a priori information on the speech signal under test [8]
LPCC	• same as LPC, but thanks to the cepstral analysis, the feature components are decorrelated [130] • increase in robustness when compared to LPC [8]	• linear scales are not adequate for the representation of speech production or perception [10] • cannot include a priori information on the speech signal under test [8]
PLP	• reduction in the discrepancy between voiced and unvoiced speech [8] • PLP peaks are reasonably independent to the length of the vocal tract [8] • resultant feature vector is low-dimensional [8] • based on short term spectrum of the speech signals [8]	• resultant feature vectors are dependent on the whole spectral balance of the formant amplitudes [8] • spectral balance is easily altered by the communication channel, noise, and the equipment used [8]
RASTA–PLP	• spectral components that change slower or quicker than the rate of change of the speech signal are suppressed [8] • robust [8, 10]	• poor performance in clean speech environments [57]
VQ	• reduction in the required memory storage size for the spectral analysis information [8] • reduction in the computational cost for the calculation of similarity between feature vectors [8] • discrete representation of speech signal [8] • fast training speed [42]	• training time increases linearly with increase in vocabulary size [42] • quantisation error in the discrete representation of speech signals [42] • temporal information is ignored [42]
PCA	• reduction in the feature vector's size, while retaining much of the significant information [131] • robust [59, 60]	• computationally expensive for high-dimensional data [8]
LDA	• maximises the distance between classes, but minimises the within class distance [132] • robust [133]	• sample distribution is assumed a priori to be Gaussian [63] • class samples are assumed to have equal variance [63]

out-of-vocabulary (OOV) words, which the ASR system will either reject or consider it as an error. The OOV words of an ASR system can be reduced by increasing the size of the training dataset. The lower the OOV rate is, the more precise the spoken language system will be [10, 122]. A recent research that tackles the problem of OOV words, for the Japanese language, is presented in [122]. In this research, a phoneme recogniser that extracts OOV words to reduce the OOV rate is presented.

As further advances are made in the area of ASR, the integration of language models and search techniques are becoming more prominent, especially when it comes to large vocabulary speech recognition applications. Owing to the various possible speech domains, the computational cost of language models increases exponentially, as the vocabulary size increases. This results from the fact that speech signals do not follow strictly a set of grammatical rules, and speaking style, regional and social dialects, need to be considered as well. Hence, a good language model needs to consider all these possibilities, but at the same time, it needs to be compact enough, for adequate real-time speech recognition [3, 123]. Owing to the various language models available, a set of criteria are required, in order to choose the optimal language model for the domain that is being considered. Some examples of such criteria are perplexity, average log likelihood, cross entropy and resultant accuracy [123, 124].

A language model defines a set of constraints on the words available in the vocabulary set, as well as their corresponding sequences. As a result, the choice of a language model determines also the resultant search space, and consequently the search technique to be used afterwards. Mainly, there are two types of language models: static and dynamic language models. A widely renowned static language technique is the *N*-gram model. In most cases, either bigrams or trigrams are considered, with the latter model performing best, since it entails more information [123]. A recent research that considered *N*-gram language models in an ASR system is presented in [23]. The language model considered estimates the probability of all possible word sequences, for a bigram language model. This system was evaluated with the TIMIT database, and the results obtained showed that with bigram language models, the phoneme accuracy was increased by approximately 3%. Another research [125], considered also the use of *N*-gram models for the reduction of OOV words.

A drawback of static language models is that such models are not capable to adapt in different domains. In this case, it is better to opt for a dynamic language model instead, since with dynamic language models the word probabilities are estimated on the speech analysed so far, and hence the model can adapt if new speech domains are considered. Examples of dynamic language models are long-distance *N*-grams, trigger-pairs, cache models and tree-based models [123].

After selecting which language model is going to be employed, a decoding search technique needs to be chosen, in order to select the best hypothesis based on a specific number of criteria. This is done by pruning those hypotheses, which have the lowest scores, through the use of a pruning algorithm [123, 126]. As a result, such methodologies are referred to as suboptimal search decisions. Two search algorithms that are widely used nowadays are the Viterbi search and *N*-best search algorithms. Starting with the Viterbi search technique, in this approach all hypotheses correspond to the same portion of speech, and hence these can be directly compared with each other.

The work presented in [127] implements an ASR system based on HMM for the task of speaker independent continuous speech recognition, with a large vocabulary. The proposed system generates a word transcription dictionary based on the word transcriptions available in the TIMIT dictionary, through the use of a dictionary compression scheme using a log-likelihood distance measure. This dictionary is then used in the Viterbi algorithm for sentence decoding. A word pair grammar is also utilised for context dependency by smoothing the transition between the words. Without the use of the dictionary and grammar information, a word accuracy of 60.1% was achieved. However, this accuracy was then further increased to 92.2%, with the addition of the dictionary and grammar information. Nonetheless, even for a medium-sized vocabulary, a complete Viterbi search is computationally expensive. Hence, a modification of the Viterbi search technique, referred to as the Viterbi beam search, is usually employed. With a beam search, only those hypotheses that fall within a certain range of the most probable hypothesis are considered [123, 128]. Further modifications for the Viterbi beam search techniques were also proposed. One such research is presented in [128], where a beam search ranking curve is identified in order to further reduce the computational time. Another recent research [129], presents an adaptive Viterbi beam search, in which the voice activity model at different stages is analysed. When compared to the conventional Viterbi beam search, an improvement in search efficiency of 35.77% was achieved.

Moving on to the second search technique, the *N*-best search can be seen as an extension of the Viterbi search technique, with the only difference that instead of choosing only the best hypothesis, with the *N*-best search approach, the *n*-best hypotheses are considered. The main drawback of the *N*-best search technique is that short hypotheses are more likely to be chosen, since longer sentences will have more errors, resulting in these hypotheses ending up with lower scores. For this reason, different modifications were proposed in order to make this search technique more efficient. Such modifications optimise either the search algorithm [123] or the pruning method that is used [126].

All of the above confirm that the advances in language processing, mostly when it comes to large vocabulary speech recognition, are of fundamental importance for the development of ASR systems and future technologies. In fact, it is believed that the ability of computers in recognising and processing speech signals as human beings will mark the arrival of truly intelligent technologies [18, 123].

2.4 Further comparison and discussion

To better understand and compare the ASR techniques discussed above, the advantages and disadvantages of the feature extraction techniques, the classification methods, and the language models are listed in Tables 1–3, respectively, along with some suggestions on the use of these techniques.

Starting with the feature extraction techniques, Table 1 shows the advantages and disadvantages of the previously discussed feature extraction approaches.

In past years, the feature extraction techniques were mainly based on cepstral analysis, such as MFCC and LPC techniques. As a matter of fact, the MFCC feature

doi: 10.1049/iet-spr.2012.0151

Table 2 Advantages and disadvantages of classification techniques

Classification technique	Advantages	Disadvantages
HMM	• able to model time distribution of speech signals [103] • simple to adapt [68] • capable to model a sequence of discrete or continuous symbols [13] • inputs can be of variable length [40]	• based on the assumption that the probability of being in a particular state is dependent only on its preceding state, ignoring any long-term dependencies [82] • emission probabilities are arbitrarily chosen; hence, these might not even represent properly the output probabilities of the corresponding state [82]
ANN (in general)	• good classifiers [16, 45] • highly adequate for pattern recognition applications [16, 45] • self-organising [16, 45] • self-learning [16, 45] • self-adaptive in new environments [16, 45] • robust [7]	• based on ERM; hence, prone to over training a local minima problems [45, 103]
MLP	• good discriminating ability [2]	• unable to model time distribution of speech signals [2] • inputs have to be of fixed length [2] • able to deal with small vocabularies only [2]
SOM	• no a priori information is required for training a SOM [134] • can easily adapt if a new sample is presented to it [134] • capable of parallel computation [134]	• SOM algorithm is not well defined mathematically; hence, values for the network parameters need to be found by trial-and-error [134] • ordered mapping obtained after the training phase may be lost when applied in real environments due to frequent adaptations [134]
RBF	• simple to implement [135] • Good discriminating ability [135] • robust [135] • online learning ability [135]	• shift invariant in time [91]
RNN	• able to model time distribution of speech signals thanks to the feedback connections [95, 103]	• complex training algorithm [94] • training algorithm is highly sensitive to any changes [94]
FNN	• does not need large amount of samples during the learning process [99] • improves convergence speed [45, 99] • not prone to local minima problems [45]	• unable to model time distribution of speech signals [45]
SVM (in general)	• Based on SRM; hence, not prone to over training and local minima problems [103] • excellent classifiers [103] • robust [103] • able to deal with high-dimensional input vectors [103]	• inputs need to be of fixed length [103] • increase in computational cost as the number of classes to be classified is increased [103] • current SVM training algorithms are not capable of dealing with huge databases [103]
one-against-all	• low number of SVM binary classifiers [101]	• large number of support vectors; hence, increase in the required storage size [101]
one-against-one	• most successful multiclass SVM method [102] • lowest computational time for training phase [107] • few support vectors [107]	• large number of SVM classifiers [101] • problem of unclassifiable regions [111]
DAGSVM	• lowest computational time for training phase [107] • fastest classification phase [107] • no unclassifiable regions [111] • few support vectors [107]	• generalisation capability is dependent on the structure of the rooted binary directed acyclic graph [111]

extraction technique was employed in numerous researches, including but not limited to continuous phoneme recognition and isolated word recognition, and it is still widely used nowadays. However, ideally the MFCC method should be employed only in clean environments because of its low robustness to noise. Apart from this, even though MFCC was also utilised in continuous speech environments, it may perform better in isolated speech environments since as stated in Table 1, one of its drawbacks is that an MFCC frame may contain information of more than one phoneme when considering continuous speech environments.

In recent years, it is becoming more obvious the need to consider also the temporal information of speech signals, and not only the frequency information. This information is not included in the MFCC and LPC techniques. As a result, feature extraction techniques based on wavelet analysis are becoming more popular, and thanks to these methods, higher accuracies are being achieved. In fact, the DWT have already proved to be superior to the well-known MFCC. Two main methods that are based on wavelet analysis are the DWT and the WPT, with the former one performing the best. These two methods are mostly employed for the task of phoneme recognition.

Another important point to keep in mind, when choosing a feature extraction technique is the amount of memory storage size available. For limited amount of storage, one must opt for a feature extraction technique that can achieve good performance with a small feature vector size, such as the DWT. However, if one wants to work with a specific extraction technique, such as the MFCC, which typically results in a 39-dimensional feature vector, one may combine either the VQ, PCA or LDA technique to the feature extraction method being used in order to reduce the dimensionality of the features extracted. The VQ, PCA and LDA techniques can also be used independently for feature extraction. However, in most cases these were used in combination with other feature extraction techniques, in order to reduce the memory storage size required while maintaining the significant information available in the features extracted. For example, the VQ approach was employed with MFCC [58], and also DWT [5, 42], in order to further improve the designed ASR system by taking advantage of the clustering property of the VQ approach. As for the PCA and LDA approaches, apart from reducing the feature vector's size, these also help in increasing the robustness of the feature extraction stage [59, 60, 133].

One last important issue to consider is whether the chosen feature extraction technique is going to be applied in a clean or a noisy environment. There are feature extraction techniques, such as the MFCC, PLP and LPC, which ideally are employed in clean speech environments, whereas other approaches, such as the DWT, WPT and LPCC, can be applied in both clean and noisy surroundings. The latter methods can also be combined with the MFCC, PLP or LPC techniques, in order to increase their robustness if these still need to be employed in noisy environments. Another approach that showed excellent performance in noisy environments is the RASTA–PLP. The RASTA–PLP proved its superiority over the PLP, MFCC and LPC methods when it comes to noisy environments. However, in clean speech environments the performance of the RASTA–PLP is very low. Hence, the RASTA–PLP should not be employed in clean speech environments.

Considering now the classification stage of an ASR system, the advantages and disadvantages of the classification techniques previously discussed are shown in Table 2. The method that is widely renowned for classification of speech signals is the HMM. The popularity of HMMs is mainly attributed to their ability in modelling the time distribution of speech signals, again confirming the importance of temporal information of speech signals. However, although huge improvements and significant recognition rates were obtained with HMMs, these are still far from achieving an optimal ASR system by themselves, due to their low generalisation capability. Hence, various modifications were proposed in order to improve the accuracies obtained by the HMMs. These modifications are mainly the consideration of ANNs and SVMs, which can be employed either independently or as hybrids with the HMMs. In the past years, ANNs were being employed more than the SVMs, with the MLP being the most popular ANN architecture in the field of ASR. Nevertheless, there were also other architectures, these being the RBF and RNN architectures, which performed better than the MLP. As a matter of fact, the RNN architecture is similar to the MLP, with the only difference of the addition of feedback connections in order to be able to model the time distribution in speech signals, which is one of the drawbacks present in an MLP architecture. As for the SOM and FNN architectures, these have also shown their potential in ASR applications, even though these were not employed extensively. However, although ANNs obtained good ASR accuracies, in recent

Table 3 Advantages and disadvantages of language model techniques

Language model technique	Advantages	Disadvantages
static	• simple to implement and powerful [123]	• unable to adapt in different domains [123] • only the very close history of the word is used [123]
dynamic	• able to adapt to new speech domains [123]	• high computational cost [123]
viterbi-beam search	• a dynamic programming technique [123] • when a principal solution is not present, a number of possible solutions are considered. On the other hand, if a clear best hypothesis exists, few other hypotheses need to be considered [123]	• if a state occurs in more than one path, the corresponding computation needs to be calculated for each path, resulting in an increase in the computational cost [123]
N-best search	• all hypotheses within the specified beam are considered [123]	• short hypotheses have a higher probability to be chosen [123]

IET Signal Process., 2013, Vol. 7, Iss. 1, pp. 25–46
doi: 10.1049/iet-spr.2012.0151

Table 4 Comparison between various ASR systems

Ref.	Year	Research work	Speaker in/ dependent (SI/SD)	Language	Feature extraction technique	Classification technique	Language model	Accuracy, %
[16]	2009	isolated word recognition	SI	Malayalam	DWT WPT	MLP	N/A	89.00 61.00
[23]	2010	context-independent phoneme recognition	SI	TIMIT Corpus – 39 classes	MFCC	CDHMM	Bigram	63.07
[26]	2011	continuous phoneme recognition	SI	TIMIT Corpus – 39 classes	MFCC	HMM-MLP	Bigram	77.83
[44]	2003	isolated word recognition	SI	English SD2 Corpus	MFCC WPT	HMM	N/A	38.77 56.90
[45]	2009	isolated word recognition	SI	50 English words	Subband MFCC	CDHMM-FNN	N/A	89.50
[51]	2011	isolated word recognition	SI	Indian	LPCC MFCC	Modified-SOM	N/A	88.05 89.27
[83]	2011	continuous phoneme recognition	SI	TIMIT Corpus – 39 classes	PLP	SMLP	N/A	78.90
[84]	2002	isolated spoken digits	SD	Urdu	MFCC	MLP	N/A	94.00
[85]	2009	isolated spoken digits	SI	Persian	MFCC & DWT	MLP	N/A	98.00
[91]	2011	isolated word recognition	SI	six English words	LPCC	RBF MLP	N/A	98.69 96.00
[92]	2009	continuous word recognition	SI	ten English words	cepstrum analysis	HMM-RBF	N/A	80.00
[110]	1999	continuous phoneme recognition	SI	TIMIT Corpus – 39 classes	MFCC	SVM	N/A	77.60
[101]	2005	word recognition	SI	DARPA RM1 Corpus	MFCC	HMM-SVM	RM word-pair grammar	94.10

years, SVMs showed their excellent classification capabilities, proving to be superior to ANNs, but most importantly researchers also showed that SVMs can achieve, either comparable, or even better results than the HMMs. As stated earlier, since the SVMs are inherently binary classifiers, a multiclass method must be employed for ASR applications. The most successful multiclass SVM method is the one-against-one method, followed by the DAGSVM.

Moving towards the last stage found in an ASR system, the advantages and disadvantages of a number of language model techniques are listed in Table 3. Nowadays, language models are becoming more essential, since more research is being carried out on large vocabulary continuous speech recognition applications. Hence, a lot of work is being done to further improve the language models and suboptimal search techniques proposed so far, with the *N*-grams and *N*-best search models being the most popular.

Concluding this review, a list of various speech recognition systems is presented in Table 4. With years, the ASR research area is focusing more on the implementation of large vocabulary continuous speaker-independent speech recognition. ASR systems are being oriented towards the implementation of hybrid models, primarily HMM–SVM hybrid architectures, in order to combine the capability of modelling the time variation present in speech signals of HMMs and the excellent classification ability of SVMs. As for the feature extraction stage, there are still numerous methods which are being employed, such as MFCC, LPCC and DWT and RASTA–PLP for noisy environments, since all of these approaches achieved good performance. Apart from this, feature extraction techniques are also being combined together, in order to benefit from the advantages of both methods. A good example is the MFDWC method [32, 41, 46], where the MFCC and DWT techniques are combined together in order to increase the robustness of the MFCC approach.

3 Conclusion

This review paper gives a brief overview of the different approaches which are widely used nowadays for the task of ASR. An ASR system is mainly composed of three components: feature extraction stage, classification stage and a language model. Various feature extraction methods were proposed, all of which achieved good performance. As regards to the classification stage, the approach which is widely used is the HMMs. Although, considerable accuracies were obtained from ASR systems based on HMMs, these are still far from achieving an optimal ASR system by themselves. Hence, numerous hybrid models, based on the concept of merging HMMs with another approach were proposed. Initially, ANNs were being employed with HMMs. However, in recent years, SVMs are also being adopted in ASR systems, where numerous researches proved the superiority of SVMs over ANNs, but most importantly researchers also showed that SVMs can achieve, either comparable or even better results than the HMMs. The last component of an ASR system is the language model. Knowledge of the language being spoken is necessary, in order to produce meaningful representation of the input speech signal. Advances in language processing are of fundamental importance for the development of ASR systems, mostly when it comes to large vocabulary speech recognition.

4 Acknowledgments

The authors thank all the anonymous reviewers for their constructive comments on an earlier version of this review paper. The research work disclosed in this publication is partially funded by the Strategic Educational Pathways Scholarship Scheme (Malta). The scholarship is part-financed by the European Union – European Social Fund.

5 References

1 Vimal Krishnan, V.R., Babu Anto, P.: 'Feature parameter extraction from wavelet subband analysis for the recognition of isolated malayalam spoken words', *Int. J. Comput. Netw. Secur.*, 2009, **1**, (1), pp. 52–55
2 Hennebert, J., Hasler, M., Dedieu, H.: 'Neural networks in speech recognition'. Sixth Microcomputer School, Prague, Czech Republic, 1994, pp. 23–40
3 Forsberg, M: 'Why is speech recognition difficult?', Chalmers University of Technology, 2003, http://www.speech.kth.se/~rolf/gslt_papers/MarkusForsberg.pdf
4 O'Shaughnessy, D.: 'Invited paper: automatic speech recognition: history, methods and challenges', *Pattern Recognit.*, 2008, **41**, (10), pp. 2965–2979
5 Ranjan, S.: 'A discrete wavelet transform based approach to Hindi speech recognition'. Int. Conf. on Signal Acquisition and Processing, 2010 (ICSAP'10), Bangalore, 2010, pp. 345–348
6 Junior, S.B., Guido, R.C., Chen, S., Vieira, L.S., Sanchez, F.L.: 'Improved dynamic time warping based on the discrete wavelet transform'. Ninth IEEE Int. Symp. Multimedia Workshops, 2007 (ISMW'07), Taichung, Taiwan, pp. 256–263
7 Vimala, C., Radha, V.: 'A review on speech recognition challenges and approaches', *World Comput. Sci. Inf. Technol.*, 2012, **2**, (1), pp. 1–7
8 Anusuya, M., Katti, S.: 'Front end analysis of speech recognition: a review', *Int. J. Speech Technol.*, 2011, **14**, (2), pp. 99–145
9 Morgan, N.: 'Deep and wide: multiple layers in automatic speech recognition', *IEEE Trans Audio Speech Lang. Process.*, 2012, **20**, (1), pp. 7–13
10 O'Shaugnessy, D.: 'Interacting with computers by voice: automatic speech recognition and synthesis', *Proc. IEEE*, 2003, **91**, (9), pp. 1272–1305
11 Rabiner, L.R., Schafer, R.W.: 'Digital processing of speech signals' (Prentice-Hall, 1978)
12 Gamulkiewicz, B., Weeks, M.: 'Wavelet based speech recognition'. 2003 IEEE 46th Midwest Symp. Circuits and Systems, Cairo, 2003, pp. 678–681
13 Mporas, I., Ganchev, T., Siafarikas, M., Fakotakis, N.: 'Comparison of speech features on the speech recognition task', *J. Comput. Sci.*, 2007, **3**, (8), pp. 608–616
14 Saha, G., Chakraborty, S., Senapati, S.: 'A new silence removal and endpoint detection algorithm for speech and speaker recognition applications'. Proc. NCC 2005, 2005
15 Zamani, B., Akbari, A., Nasersharif, B., Jalalvand, A.: 'Optimised discriminative transformations for speech features based on minimum classification error', *Pattern Recognit. Lett.*, 2011, **32**, (7), pp. 948–955
16 Vimal Krishnan, V.R., Babu Anto, P.: 'Features of wavelet packet decomposition and discrete wavelet transform for malayalam speech recognition', *Recent Trends Eng.*, 2009, **1**, (2), pp. 93–96
17 Alkhaldi, W., Fakhr, W., Hamdy, N.: 'Automatic speech recognition in noisy environments using wavelet transform'. 2002. Available from: http://www.wseas.us/e-library/conferences/skiathos2002/papers/447-231.pdf
18 Jurafsky, D., Martin, J.H.: 'Speech and language processing' (Prentice-Hall, 2009)
19 Liddy, E.D.: 'Natural language processing in encyclopedia of library and information science' (Marcel Decker, Inc., NY, 2001, 2nd edn.)
20 Leung, K.F., Leung, F.H.F., Lam, H.K., Tam, P.K.S.: 'Recognition of speech commands using a modified neural fuzzy network and an improved GA'. 12th IEEE Int. Conf. on Fuzzy Systems, 2003, (FUZZ'03), Kowloon, China, 2003, pp. 190–195
21 Lasserre, J., Bishop, C.M.: 'Generative or Discriminative? Getting the best of both worlds'. Bayesian Statistics, vol. 8. Microsoft Research, 2007
22 Du, X.P., He, P.L.: 'The clustering solution of speech recognition models with SOM'. Lecture Notes in Computer Science. Advances in Neural Networks – ISNN 2006 (Springer Berlin/Heidelberg, 2006), pp. 150–157
23 Ben Messaoud, Z., Ben Hamida, A.: 'CDHMM parameters selection for speaker-independent phone recognition in continuous speech system'. MELECON 2010 – 2010 15th IEEE Mediterranean Electrotechnical Conf., Valletta, 2010, pp. 253–258
24 Korba, M.C.A., Messadeg, D., Djemili, R.H.B.: 'Robust speech recognition using perceptual wavelet denoising and mel-frequency product spectrum cepstral coefficient features', *Informatica*, 2008, **32**, pp. 283–288
25 Nouza, J., Zdansky, J., Cerva, P.: 'System for automatic collection, annotation and indexing of Czech broadcast speech with full-text search'. MELECON 2010 – 2010 15th IEEE Mediterranean Electrotechnical Conf., Valletta, 2010, pp. 202–205
26 Toth, L.: 'A hierarchical, context-dependent neural network architecture for improved phone recognition'. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2011, Prague, 2011, pp. 5040–5043
27 Muller, D.N., de Siqueira, M.L., Navaux, P.O.A.: 'A connectionist approach to speech understanding'. Int. Joint Conf. on Neural Networks, 2006 (IJCNN'06), Vancouver, BC, 2006, pp. 3790–3797
28 Smaragdis, P., Radhakrishnan, R., Wilson, K.W.: 'Content extraction through audio signal analysis', in Divakaran, A., (Ed.): 'Multimedia content analysis' (Springer, 2009), pp. 1–34
29 Wicks, M.A.: 'The mel frequency scale and coefficients'. 1998. Available from: http://kom.aau.dk/group/04gr742/pdf/MFCC_worksheet.pdf
30 Hung, J.-W., Fan, H.-T.: 'Subband feature statistics normalisation techniques based on a discrete wavelet transform for robust speech recognition', *IEEE Signal Process. Lett.*, 2009, **16**, (9), pp. 806–809
31 Gupta, M., Gilbert, A.: 'Robust speech recognition using wavelet coefficient features'. IEEE Workshop on Automatic Speech Recognition and Understanding, 2001 (ASRU '01), 2001, pp. 445–448
32 Xuefei, L.: 'A new wavelet threshold denoising algorithm in speech recognition'. Asia-Pacific Conf. on Information Processing, 2009 (APCIP 2009), Shenzhen, 2009, pp. 310–313
33 Nehe, N.S., Holambe, R.S.: 'New feature extraction techniques for Marathi digit recognition', *Int. J. Recent Trends Eng.*, 2009, **2**, (2), pp. 22–24
34 Polikar, R.: 'The wavelet tutorial'. 1996. Available from: http://users.rowan.edu/~polikar/wavelets/wttutorial.html
35 Mallat, S.G.: 'A theory for multiresolution signal decomposition: the wavelet representation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1989, **11**, (7), pp. 674–693
36 Sudhakar: 'The discrete wavelet transform'. 2003. Available from: http://etd.lib.fsu.edu/theses/available/etd-11242003-185039/unrestricted/09_ds_chapter2.pdf
37 Vetterli, M., Herley, C.: 'Wavelets and filter banks: relationships and new results'. 1990 Int. Conf. on Acoustics, Speech, and Signal Processing, 1990 (ICASSP'90), Albuquerque, NM, USA, 1990, pp. 1723–1726
38 Hunt, A., Favero, R.: 'Using principal component analysis with wavelets in speech recognition'. SST Conf., ASSTA Inc., Perth, 1994, pp. 296–301
39 Walker, S.L., Foo, S.Y.: 'Optimal wavelets for speech signal representations', *Syst. Cybern. Inf.*, 2003, **1**, (4), pp. 44–46
40 Milone, D.H., Di Persia, L.E.: 'Learning hidden Markov models with hidden Markov trees as observation distributions'. Ninth Argentine Symp. Artificial Intelligence (ASAI 2007), Mar del Plata, Argentina, 2007, pp. 13–22
41 Tavanaei, A., Manzuri, M.T., Sameti, H.: 'Mel-scaled discrete wavelet transform and dynamic features for the Persian phoneme recognition'. Int. Symp. Artificial Intelligence and Signal Processing (AISP), 2011, Tehran, 2011, pp. 138–140
42 Krishnan, M., Neophytou, C.P., Prescott, G.: 'Wavelet transform speech recognition using vector quantisation, dynamic time warping and artificial neural networks', Computer Aided Systems Engineering and Telecommunications & Information Science Laboratory, 1994
43 Tan, B.T., Fu, M., Spray, A., Dermody, P.: 'The use of wavelet transforms in phoneme recognition'. Proc. Fourth Int. Conf. on Spoken Language, 1996 (ICSLP'96), Philadelphia, PA, USA, 1996, pp. 2431–2434
44 Modic, R., Lindberg, B., Petek, B.: 'Comparative wavelet and MFCC speech recognition experiments on the Slovenian and English SpeechDat2'. Proc. ISCA Tutorial and Research Workshop on Non-Linear Speech Processing, Denmark, 2003
45 Zhou, P., Tang, L.Z., Xu, D.F.: 'Speech recognition algorithm of parallel subband HMM based on wavelet analysis and neural network', *Inf. Technol. J.*, 2009, **8**, pp. 796–800
46 Tufekci, Z., Gowdy, J.N., Gurbuz, S., Patterson, E.: 'Applied mel-frequency discrete wavelet coefficients and parallel model compensation for noise-robust speech recognition', *Speech Commun. Sci. Direct*, 2006, **48**, pp. 1294–1307
47 Thiang, Wijoyo, S.: 'Speech recognition using linear predictive coding and artificial neural network for controlling movement of mobile robot'. Int. Conf. on Information and Electronics Engineering, Singapore, 2011, pp. 179–183
48 Bradbury, J.: 'Linear predictive coding'. 2000. Available from: http://my.fit.edu/~vKepuska/ece5525/lpc_paper.pdf
49 Nataraj, K.S., Jagbandhu, J., Pandey, P.C., Shah, M.S.: 'Improving the consistency of vocal tract shape estimation'. National Conf. on Communications (NCC), 2011, Bangalore, 2011, pp. 1–5
50 Cheng, O., Abdulla, W., Salcic, Z.: 'Performance evaluation of front-end processing for speech recognition systems'. School of

doi: 10.1049/iet-spr.2012.0151

Engineering Report. The University of Auckland, Electrical and Computer Engineering, 2005. Report No. 621.
51 Venkateswarlu, R.L.K., Kumari, R.V.: 'Novel approach for speech recgonition by using Self-Organised Maps'. 2011 Int. Conf. on Emerging Trends in Networks and Computer Communications (ETNCC), Udaipur, 2011, pp. 215–222
52 Li, T.F., Chang, S.C.: 'Speech recognition of mandarin syllables using both linear predict coding cepstra and Mel frequency cepstra'. Proc. 19th Conf. on Computational Linguistics and Speech Processing, Taiwan, 2007
53 Yusof, Z., Ahmed, M.: '2009. Available from: http://rps.bmi.unikl.edu.my/jnp/archive/2009/2009-197.pdf
54 Ganapathy, S., Thomas, S., Hermansky, H.: 'Modulation frequency features for phoneme recognition in noisy speech', *J. Acoust. Soc. Am.*, 2009, **125**, pp. EL8–EL12
55 Sarosi, G., Mozsary, M., Mihajlik, P., Fegyo, T.: 'Comparison of feature extraction methods for speech recognition in noise-free and in traffic noise environment'. Sixth Conf. on Speech Technology and Human-Computer Dialogue (SpeD), 2011, Brasov, 2011, pp. 1–8
56 Hermansky, H., Morgan, N., Bayya, A., Kohn, P.: 'RASTA-PLP speech analysis'. ICSI Technology Report. Internation Computer Science Institute, Berkeley, CA, 1991. Report No.: TR-91-069
57 Anusuya, M.A., Katti, S.K.: 'Comparison of different speech feature extraction techniques with and without wavelet transform to Kannada speech recognition', *Int. J. Comput. Appl.*, 2011, **26**, (4), pp. 19–23
58 Hu, X., Zhan, L., Xue, Y., Zhou, W., Zhang, L.: 'Spoken arabic digits recognition based on wavelet neural networks'. 2011 IEEE Int. Conf. on Systems, Man and Cybernetics (SMC), Anchorage, AK, 2011, pp. 1481–1485
59 Veisi, H., Sameti, H.: 'The integration of principal component analysis and cepstral mean subtraction in parallel model combination for robust speech recognition', *Digit. Signal Process.*, 2011, **21**, (1), pp. 36–53
60 Lee, J.Y., Hung, J.: 'Exploiting principal component analysis in modulation spectrum enhancement for robust speech recognition'. 2011 Eighth Int. Conf. on Fuzzy Systems and Knowledge Discovery (FSKD), Shanghai, 2011, pp. 1947–1951
61 Takiguchi, T., Ariki, Y.: 'PCA-based speech enhancement for distorted speech recognition', *J. Multimedia*, 2007, **2**, pp. 13–18
62 Viszlay, P., Juhaar, J., Pleva, M.: 'Alternative phonetic class definition in linear discriminant analysis of speech'. 19th Int. Conf. on Systems, Signals and Image Processing (IWSSIP), 2012, Vienna, 2012, pp. 655–658
63 Garau, G., Renals, S.: 'Combining spectral representations for large vocabulary continuous speech recognition', *IEEE Trans. Audio Speech Language Process.*, 2008, **16**, (3), pp. 508–518
64 Ben Messaoud, Z., Ben Hamida, A.: 'Combining formant frequency based on variable order LPC coding with acoustic features for TIMIT phone recognition', *Int. J. Speech Technol.*, 2011, **14**, pp. 393–403
65 Paulson, L.D.: 'Speech recognition moves from software to hardware', *Computer*, 2006, **39**, (11), pp. 15–18
66 Lazli, L., Sellami, M.: 'Connectionist probability estimators in HMM Arabic speech recognition using fuzzy logic'. Proc. MLDM, 2003, pp. 379–388
67 Birkenes, Ø., Matsui, T., Tanabe, K., Siniscalchi, S.M., Myrvoll, T.A., Johnsen, M.H.: 'Penalised logistic regression with HMM log-likelihood regressors for speech recognition', *IEEE Trans. Audio Speech Language Process.*, 2010, **18**, (6), pp. 1440–1454
68 Juang, B.H., Rabiner, L.R.: 'Hidden Markov models for speech recognition', *Technometrics*, 1991, **33**, (3), pp. 251–272
69 Nguyen, P., Heigold, G., Zweig, G.: 'Speech recognition with flat direct models', *Sel. Topics Signal Process. IEEE J.*, 2010, **4**, (6), pp. 994–1006
70 Abdulla, W.H., Kasabov, N.: 'The concepts of hidden Markov model in speech recognition' (University of Otago, 1999)
71 Rabiner, L.: 'A tutorial on HMM and selected applications in speech recognition', *Proc. IEEE*, 1989, **77**, (2), pp. 257–286
72 Lee, K.F.H.H.W.: 'Speaker-independent phone recognition using hidden Markov models', *IEEE Trans. Acoust. Speech Signal Process.*, 1989, **37**, (11), pp. 1641–1648
73 Ketabdar, H., Bourlard, H.: 'Enhanced phone posteriors for improving speech recognition systems', *IEEE Trans. Audio Speech Lang. Process.*, 2010, **18**, (6), pp. 1094–1106
74 Hermansky, H., Ellis, D.P.W., Sharma, S.: 'Tandem connectionist feature extraction for conventional HMM systems'. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 2000 (ICASSP'00), Istanbul, Turkey, 2000, pp. 1635–1638
75 Crouse, M.S., Nowak, R.D., Baraniuk, R.G.: 'Wavelet-based statistical signal processing using hidden Markov models', *IEEE Trans. Signal Process.*, 1998, **46**, (4), pp. 886–902
76 Jung, S., Son, J., Bae, K.: 'Feature extraction based on wavelet domain hidden Markov tree model for robust speech recognition'. AI 2004: Advances in Artificial Intelligence, (Springer, Berlin/Heidelberg, 2004), pp. 1154–1159
77 Chang, T.H., Luo, Z.Q., Deng, L., Chi, C.Y.: 'A convex optimisation method for joint mean and variance parameter estimation of large-margin CDHMM'. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 2008 (ICASSP 2008), Las Vegas, NV, 2008, pp. 4053–4056
78 Young, S., Evermann, G.M.G., Hain, T., Kershaw, D.: 'HTK - Hidden Markov Model Toolkit (Ver 3.4)'. 2006. Available from: http://htk.eng.cam.ac.uk/
79 Jiang, H., Li, X., Liu, C.: 'Large margin hidden Markov models for speech recognition', *IEEE Trans. Audio Speech Language Process.*, 2006, **14**, (5), pp. 1584–1595
80 Sha, F., Saul, L.K.: 'Large margin hidden Markov models for automatic speech recognition', *Adv. Neural Inf. Process. Syst.*, 2007, **1**, pp. 1249–1256
81 Chen, J.C., Chien, J.T.: 'Bayesian large margin hidden Markov models for speech recognition'. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 2009 (ICASSP 2009), Taipei, 2009, pp. 3765–3768
82 Trentin, E., Gori, M.: 'Robust combination of neural networks and hidden Markov models for speech recognition', *IEEE Trans. Neural Netw.*, 2003, **14**, (6), pp. 1519–1531
83 Sivaram, G.S.V.S., Hermansky, H.: 'Multilayer perceptron with sparse hidden outputs for phoneme recognition'. 2011 IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP), Prague, 2011, pp. 5336–5339
84 Ahad, A., Fayyaz, A., Mehmood, T.: 'Speech recognition using multilayer perceptron'. IEEE Proc. Students Conf., 2002 (ISCON'02), 2002, pp. 103–109
85 Pour, M.M., Farokhi, F.: 'A new approach for Persian speech recognition'. IEEE Int. Advance Computing Conf., 2009 (IACC 2009), Patiala, 2009, pp. 153–158
86 Sivaram, G.S.V.S., Hermansky, H.: 'Sparse multilayer perceptron for phoneme recognition', *IEEE Trans. Audio, Speech Lang. Process.*, 2012, **20**, (1), pp. 23–29
87 Cutajar, M., Gatt, E., Micallef, J., Grech, I., Casha, O.: 'Digital hardware implementation of Self-Organising Maps'. 15th IEEE Mediterranean Electrotechnical Conf. MELECON 2010 – 2010, Valletta, 2010, pp. 1123–1128
88 Cutajar, M., Gatt, E.: 'Digital implementation of Self-Organising Maps. Final year project, Faculty of Engineering, Department of Microelectronics Engineering, Malta, 2009
89 Campos, M.M., Carpenter, G.A.: 'WSOM: building adaptive wavelets with self-organizing maps'. IEEE World Congress on Computational Intelligence. The 1998 IEEE Int. Joint Conf. on Neural Networks Proc., 1998., Anchorage, AK, USA, 1998, pp. 763–767
90 Paul, A.K., Das, D., Kamal, M.M.: 'Bangla speech recognition system using LPC and ANN'. Seventh Int. Conf. on Advances in Pattern Recognition, 2009 (ICAPR'09), Kolkata, 2009, pp. 171–174
91 Venkateswarlu, R.L.K., Kumari, R.V., Jayasri, G.V.: 'Speech recognition using radial basis function neural network'. Third Int. Conf. on Electronics Computer Technology (ICECT), 2011, Kanyakumari, 2011, pp. 441–445
92 Umarani, S.D., Raviram, P., Wahidabanu, R.S.D.: 'Implementation of HMM and radial basis function for speech recognition'. Int. Conf. on Intelligent Agent and Multi-Agent Systems, 2009 (IAMA 2009), Chennai, 2009, pp. 1–4
93 Hou, X.: 'Noise robust speech recognition based on wavelet-RBF neural network'. Poc. SPIE, 2009, vol. 7490
94 Veera, A.K.: 'Speech recognition based on artificial neural networks'. 2004. Available from: http://www.cis.hut.fi/Opinnot/T-61.6040/pellom-2004/project-reports/project_07.pdf
95 Koizumi, T., Mori, M., Taniguchi, S., Maruya, M.: 'Recurrent neural networks for phoneme recognition'. Proc. Fourth Int. Conf. on Spoken Language, 1996 (ICSLP'96), Philadelphia, 1996, pp. 326–329
96 Vinyals, O., Ravuri, S.V., Povey, D.: 'Revisiting recurrent neural networks for robust ASR'. 2012 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2012, Kyoto, pp. 4085–4088
97 Uma Maheswari, N., Kabilan, A.P., Venkatesh, R.: 'Speaker independent phoneme recognition using neural networks', *J. Theoret. Appl. Inf. Technol.*, 2009, **6**(2), pp. 230–235
98 Helmi, N., Helmi, B.H.: 'Speech recognition with fuzzy neural network for discrete words'. 2008 Fourth Int. Conf. on Natural Computation, 2008, pp. 265–269
99 Sabah, R., Aino, R.N.: 'Isolated digit speech recognition in Malay language using neuro-fuzzy approach'. 2009 Third Asia Int. Conf. on Modelling and Simulation, 2009, pp. 336–340

100 Tang, H., Meng, C.H., Lee, L.S.: 'An initial attempt for phoneme recognition using Structured Support Vector Machine (SVM)'. 2010 IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP), Dallas, TX, 2010, pp. 4926–4929

101 Kruger, S.E., Schaffoner, M., Katz, M., Andelic, E., Wendemuth, A.: 'Speech recognition with support vector machines in a hybrid system'. Proc. EuroSpeech 2005, 2005

102 Sonkamble, B.A., Doye, D.D., Sonkamble, S.: 'An efficient use of support vector machines for speech signal signal classification'. Proc. Eighth WSEAS Int. Conf. Computational Intelligence, Man–Mmachine Systems and Cybernetics, 2009, pp. 117–120

103 Solera-Urena, R., Padrell-Sendra, J., Martin-Iglesias, D., Gallardo-Antolin, A., Pelaez-Moreno, C., Diaz-De-Maria, F.: 'SVMs for automatic speech recognition: a survey', Progress in nonlinear speech processing (Springer-Verlag, Berlin, Heidelberg, 2007), pp. 190–216

104 Haykin, S.: 'Neural networks: a comprehensive foundation' (Prentice-Hall, 1999)

105 Weston, J., Watkins, C.: 'Support vector machines for multiclass pattern recognition'. Proc. Seventh European Symp. Artificial Neural Networks, 1999, pp. 219–224

106 Franc, V., Hlavac, V.: 'Multi-class support vector machine'. Proc. ICPR, Quebec, 2002, pp. 236–239

107 Hsu, C.W., Lin, C.J.: 'A comparison of methods for multiclass support vector machines', *IEEE Trans. Neural Netw.*, 2002, **13**, (2), pp. 415–425

108 Duan, K., Keerthi, S.S.: 'Which is the best multiclass SVM method? an empirical study'. Proc. Multiplie Classifier Systems, 2005, pp. 278–285

109 Hastie, T., Tibshirani, R.: 'Classification by pairwise coupling', *Annal. Stat.*, 1998, **26**, (2), pp. 451–471

110 Clarkson, P., Moreno, P.J.: 'On the use of support vector machines for phonetic classification'. Proc. 1999 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 1999, Phoenix, AZ, USA, 1999, pp. 585–588

111 Abe, S.: 'Analysis of multiclass support vector machines'. Proc. Int. Conf. on Computational Intelligence for Modelling, Control and Automation (CIMCA 2003), Vienna, Austria, 2003, pp. 385–396

112 Tsujinishi, D., Koshiba, Y., Abe, S.: 'Why pairwise is better than one-against-all or all-at-once'. Proc. 2004 IEEE Int. Joint Conf. on Neural Networks, 2004, 2004

113 Venkataramani, V., Chakrabartty, S., Byrne, W.: 'Ginisupport vector machines for segmental minimum Bayes risk decoding of continuous speech', *Comput. Speech Lang.*, 2007, **21**, (3), pp. 423–442

114 Ganapathiraju, A., Hamaker, J.E., Picone, J.: 'Applications of support vector machines to speech recognition', *IEEE Trans. Signal Process.*, 2004, **52**, (8), pp. 2348–2355

115 Xiao-feng, L., Xue-ying, Z., Ji-kang, D.: 'Speech recognition based on support vector machine and error correcting output codes'. 2010 First Int. Conf. on Pervasive Computing Signal Processing and Applications (PCSPA), Harbin, 2010, pp. 336–339

116 Thubthong, N., Kijsirikul, B.: 'Support vector machines for Thai phoneme recognition', *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 2001, **9**, (6), pp. 803–813

117 Li, J.: 'An empirical comparison between SVMs and ANNs for speech recognition'. The First Instructional Conf. on Machine Learning, iCML-2003, 2003

118 Toth, L., Kocsor, A.: 'Application of kernel-based feature space transformations and learning methods to phoneme classification', *Appl. Intell.*, 2004, **21**, (2), pp. 129–142

119 García Moral, A.I., Solera Ureña, R., Peláez-Moreno, C., Díaz-de-María, F.: 'Hybrid models for automatic speech recognition: a comparison of classical ANN and kernel based methods'. (Springer, 2007, *LNCS*), pp. 51–54

120 Jamieson, K., Gupta, M.R., Swanson, E., Anderson, H.S.: 'Training a support vector machine to classify signals in a real environment given clean training data'. 2010 IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP), Dallas, TX, 2010, pp. 2214–2217

121 Dowding, J.: 'Reducing search by partitioning the word network'. Proc. Workshop on Speech and Natural Language, 1989

122 Kotwal, M.R.A., Hassan, F., Muhammad, G., Huda, M.N.: 'Tandem MLNs based phonetic feature extraction for phoneme recognition', *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, 2011, **3**, pp. 88–95

123 Deshmukh, N., Picone, J.: 'Methodologies for language modeling and search in continuous speech recognition'. Proc. IEEE Southeastcon'95. Visualize the Future, Raleigh, NC, 1995, pp. 192–198

124 Rosenfeld, R.: 'Two decades of statistical language modeling: where do we go from here?', *Proc. IEEE*, 2000, **88**, (8), pp. 1270–1278

125 Lecorvé, G., Gravier, G., Sébillot, P.: 'Automatically finding semantically consistent n-grams to add new words in LVCSR systems'. Proc. ICASSP 2011, 2011, pp. 4676–4679

126 Illina, I., Gong, Y.: 'Improvement in N-best search for continuous speech recognition'. Proc. Fourth Int. Conf. on Spoken Language, 1996 (ICSLP'96), 1996, pp. 2147–2150

127 Zhao, Y., Wakita, H., Zhuang, X.: 'An HMM based speaker-independent continuous speech recognition system with experiments on the TIMIT database'. 1991 Int. Conf. on Acoustics Speech and Signal Processing, 1991 (ICASSP-91), Toronto, ON, 1991, pp. 333–336

128 Jang, J., Lin, S.: 'Optimisation of Viterbi beam search in speech recognition'. Int. Symp. Chinese Spoken Language Processing, 2002

129 Wei, L., Weisheng, H.: 'Improved Viterbi algorithm in continuous speech recognition'. 2010 Int. Conf. on Computer Application and System Modeling (ICCASM), Taiyuan, 2010, pp. 207–209

130 Kesarkar, M.P.: 'Feature extraction for speech recognition'. M.Tech. Credit Seminar Report. Electronic Systems Group, EE. Department, IIT, Bombay, 2003

131 Thatphithakkul, N., Kruatrachue, B., Wutiwiwatchai, C., Marukatat, S., Boonpiam, V.: 'Robust speech recognition using pca-based noise classification'. SPECOM, 2005 October, p. 2548

132 Dengfeng, K., Shuang, X., Bo, X.: 'Optimization of tone recognition via applying linear discriminant analysis in feature extraction'. 2008 Third Int. Conf. on Innovative Computing Information and Control (ICICIC), Dalian, Liaoning China, 2008, pp. 528–531

133 Fontaine, V., Ris, C., Leich, H.: 'Nonlinear discriminant analysis with neural networks for speech recognition'. Proc. EUSIPCO 96, EURASIP 1996, pp. 1583–1586

134 Sayers, C.: 'Self Organising Feature Maps and their Applications to Robotics'. Technical Reports (CIS). Department of Computer and Information Science, School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, 1991. Report No.: MS-CIS-91-46

135 Hao, Y., Tiantian, X., Paszczynski, S., Wilamowski, B.M.: 'Advantages of radial basis function networks for dynamic system design', *IEEE Trans. Ind. Electron.*, 2011, **58**, (12), pp. 5438–5450

doi: 10.1049/iet-spr.2012.0151