Dear Author,

Please, note that changes made to the HTML content will be added to the article before publication, but are not reflected in this PDF.

Note also that this file should not be used for submitting corrections.
An overview of applications and advancements in automatic sound recognition

Roneel V. Sharan *, Tom J. Moir

School of Engineering, Auckland University of Technology, Private Bag 92006, Auckland 1142, New Zealand

Abstract

Automatic sound recognition (ASR) has attracted increased and wide ranging interests in recent years. In this paper, we carry out a review of some important contributions in ASR techniques, mainly over the last one and a half decades. Similar to speech recognition systems, the robustness of an ASR system largely depends on the choice of feature(s) and classifier(s). We take a wider perspective in providing an overview of the features and classifiers used in ASR systems starting from early works in content-based audio classification to more recent developments in applications such as sound event recognition, audio surveillance, and environmental sound recognition. We also review techniques that have been utilized in noise robust sound recognition systems and feature optimization methods. Finally, some of the less commonly known applications of ASR are discussed.

© 2016 Published by Elsevier B.V.

1. Introduction

Any given environment generally contains a number of different sounds. In early literature, these sounds were often divided into speech and non-speech. The task of non-speech sound classification is now more commonly known as automatic sound recognition (ASR). It is also referred as sound event recognition (SER) and acoustic event detection in some contexts. ASR systems are designed to recognize sounds automatically using signal processing and machine learning techniques. In concept, it is very similar to an automatic speech recognition system except that the input signal is non-speech. Research in speech recognition has received significant attention over the past few decades, whereas in ASR only seems to have intensified over the past two decades or so.

There are many applications of an ASR system. Initial interests in ASR were mostly centered around content-based audio classification and retrieval [1–3] and speech and non-speech recognition [4,5]. The specific application of most of these initial works were unclear but were eventually streamlined into applications such as music genre classification [6] and musical instrument sound classification [7].

However, applications have diversified since then with interests in areas such as audio surveillance [8], sound event recognition [9], and environmental sound recognition [10]. While audio surveillance can be seen as a subclass of sound event recognition, there are some differences as discussed in Section 4.1. Applications of audio surveillance and sound event recognition systems include security monitoring in a room [11] and public transport [12], intrusion detection in wildlife areas [13], and monitoring of elderly people, also referred as medical telemonitoring [14]. Environmental sound recognition poses a greater challenge when compared to audio surveillance and sound event recognition applications. This is because an environmental sound can comprise a number of different sound events within the environment which can be present in different combinations at any given time.

While the applications of ASR are many, the general approach to these pattern recognition problems are same and generally inspired from techniques employed in speech recognition systems. An overview of a statistical pattern classifier adopted in most ASR systems is given in Fig. 1. The three key steps in implementing an ASR system are signal preprocessing, feature extraction, and classification. Signal preprocessing seeks to prepare the signal for feature extraction. Typically, a signal is divided into smaller frames, often in the range of 10–30 ms, and a window function is applied to smooth the signal for further analysis. Hamming window seems to be the preferred choice in most ASR systems. While speech recognition systems typically use a sampling frequency of 8000 Hz, ASR systems employ a sampling frequency of 8000 Hz or higher, common values are 16000 Hz, 22050 Hz, and 44100 Hz, largely depending on the frequency bands of the sound signals.

* Corresponding author.
E-mail address: roneel.sharan@aut.ac.nz (R.V. Sharan).
http://dx.doi.org/10.1016/j.neucom.2016.03.020
0925-2312/© 2016 Published by Elsevier B.V.

Please cite this article as: R.V. Sharan, T.J. Moir, An overview of applications and advancements in automatic sound recognition, Neurocomputing (2016), http://dx.doi.org/10.1016/j.neucom.2016.03.020
considered in the application. Depending on the sampling frequency of the signal, a frame size of 256, 512, or 1024 samples are normally chosen with some degree of overlap between adjacent frames, such as 25% or 50%, to prevent loss of information around the edges of the window. Inherent features are then extracted from the signal and the input signal is represented by a feature vector in a much simpler and condensed form, which is referred as feature extraction. The time domain signal is often transformed to frequency domain or time-frequency domain for this purpose. Based on a set of training data containing observations whose classes are known, the task of the classifier is then to assign unknown observations to one of the classes.

Features and classifiers from speech recognition systems are often employed in ASR systems. While most of the traditional features continue to be used today, they are often complemented with new features for improved performance. A thorough review of features for audio classification is provided in [15] where the features are distinguished based on its domain and can be summarized as:

- **Temporal domain** – based on the aspect of the signal the feature represents such as amplitude, power, and zero-crossing rate.
- **Frequency domain** – which can be further divided into perceptual features, which have a semantic meaning to the human listener, and physical features, which give description in terms of mathematical, statistical, and physical properties of the audio signal.
- **Cepstral features** – approximate the spectral envelope.
- **Modulation frequency features** – provide information on long-term amplitude or frequency variation of the signal.
- **Eigen domain features** – representing long-term information contained in sound segments with duration of several seconds.
- **Phase space features** – which capture information orthogonal to features originating from linear models.

Time domain, frequency domain, and cepstral features are by far the most commonly used features in ASR systems.

In addition, the commonly used classifiers are k-nearest neighbor (kNN), Gaussian mixture model (GMM), hidden Markov model (HMM), artificial neural networks (ANN), and support vector machines (SVMs), which have been well defined in many literature. While all of these continue to be used today, modifications and hybrid classification algorithms have been proposed over the years. Also, deep learning methods, such as deep neural networks (DNNs), have gained significant attention in various pattern recognition problems in recent years.

The classification performance of an ASR system is mostly reported using the classification accuracy which can be given in percentage as number of correctly classified test samples divided by the total number of test samples. The error rate (ER) can also be used for this purpose which can be stated as the number of misclassified test samples divided by the total number of test samples.

In this work, we focus on the features and classifiers used in ASR systems as seen through applications in content-based audio classification and retrieval, audio surveillance, sound event recognition, and environmental sound recognition. Due to its widespread usage in ASR systems and being a relatively new classifier, we put particular emphasis on SVMs. We also give an overview of DNNs which has seen an increased usage in speech recognition systems recently and employed in some ASR systems as well. In addition, we review literature in noise robust sound recognition and feature optimization methods, and discuss some of the lesser known applications of ASR.

### 2. Feature extraction muscle

#### 2.1. Time and frequency domain features

One of the early works in content-based audio classification and retrieval is by Wold et al. [1] which also found commercial success and was called Muscle Fish (www.musclefish.com). It utilized some low-level acoustical features, such as loudness, pitch, brightness or spectral centroid (SC), and bandwidth (BW), with a nearest neighbor (NN) classifier based on normalized Euclidean distance. The sound database had 409 sounds files belonging to 16 classes: alto trombone, animals, bells, cello bowed, crowds, female, laughter, machines, male, oboe, percussion, telephone, tubular bells, violin bowed, violin pizz, and water. Content-based retrieval has been the main application of this work with Virage Inc., BBC, and Kodak amongst its licensees [16].

Some other commonly used time and frequency domain features include zero-crossing rate (ZCR), short-time energy (STE), subband energy (SBE), spectral flux (SF), and spectral roll-off (SR). While time and frequency domain features continue to be used in ASR systems, such as in audio surveillance applications [8,17], they are often only used as supplementary features.

#### 2.2. Cepstral features

##### 2.2.1. MFCCs

Li [2] used the Muscle Fish database and improved the ER when compared to [1] with the introduction of mel-frequency cepstral coefficients (MFCCs) as features and the nearest feature line (NFL) [18] method of classification. Humans are better at differentiating small changes in pitch at low frequencies than at high frequencies. MFCCs equally space the frequency bands on the mel-scale which more closely resembles how humans perceive sound when compared to linearly spaced cepstrums. In addition, a cepstrum gives...
information about how the frequencies change in different spectrum bands. Therefore, a combination of mel-scale and cepstrum make MFCCs useful in audio classification.

In computation of MFCCs, firstly, the discrete Fourier transform (DFT) is applied to the windowed signal as

\[ X(k, t) = \sum_{n=0}^{N} x(n)w(n)e^{-j2\pi nk/N}, \quad k = 0, \ldots, N-1 \]  

where \( N \) is the length of the window, \( x(n) \) is the time-domain signal, \( X(k, t) \) is the \( k \)th harmonic corresponding to the frequency \( f(k) = kf_t/N \) for the \( t \)th frame, \( F_t \) is the sampling frequency, and \( w(n) \) is the window function.

The conversion from frequency in Hz, \( f_{Hz} \), to frequency in mel, \( f_{Mel} \), can be given as [19]

\[ f_{Mel} = 1127 \log \left(1 + \frac{f_{Hz}}{700}\right) \]  

The center frequency for the \( m \)th filter can be computed as

\[ f_{cm} = f_l + \frac{m(f_h - f_l)}{M - 1} \quad m = 1, 2, \ldots, M \]  

where all the frequency values are given in mel, \( f_l \) and \( f_h \) are the minimum and maximum cut-off frequencies, respectively, and \( M \) is the total number of mel-filters. The adjacent filters overlap such that the lower and upper end of a filter are located at the center frequency of the previous and next filter, respectively, while the peak of the filter is at its center frequency.

For the \( t \)th frame, the output of the \( m \)th filter, referred as filter bank energies, can then be determined as

\[ E(m, t) = \sum_{k=0}^{N-1} V(m, k)|X(k, t)|^2 \quad m = 1, 2, \ldots, M \]  

where \( V(m, k) \) is the normalized frequency response.

The MFCCs are obtained as the discrete cosine transform (DCT) of the log compressed filter bank energies given as

\[ c(i, t) = \left(\frac{2}{M_i}\right) \sum_{m=1}^{M_i} \log(E(m, t)) \cos\left(\frac{\pi}{M_i}(m - 0.5)\right), \quad i = 1, 2, \ldots, L \]  

where \( L \) is the order of the cepstrum.

The number of filters in the bank is normally in the range of 20–24 with 19 filters used in [2], 20 filters in [8,20], 23 filters in [10,17] and 24 filters in [9]. In addition, the first and second derivatives of the coefficients, commonly known as delta and delta-delta coefficients, respectively, which provide trajectories of MFCCs over time, are often appended to improve the classification performance of ASR systems. The delta coefficients can be computed as [21]

\[ c_{\delta i}(t) = \frac{1}{2} \sum_{d=1}^{D} d[c(i+d, t) - c(i-d, t)] \]  

where \( c_{\delta i}(t) \) is the \( i \)th delta coefficient in the \( t \)th frame and the value of \( D \) is often set to 2. The same formula can be applied to the delta coefficients to compute the delta-delta coefficients, \( c_{\delta \delta i}(t) \).

The feature vector for each sound signal is often represented by the mean and standard deviation values along each feature dimension. However, to reduce the effect of different environmental conditions, the coefficients are often normalized before feature vector formation. Cepstral mean and variance normalization (CMVN) is the most common data normalization method which can be given as

\[ c_{CMVN}(i, t) = \left(\frac{c(i, t) - \mu(i)}{\sigma(i)}\right) \]  

where \( \mu(i) \) and \( \sigma(i) \) are the mean and variance along the \( i \)th dimension, respectively. Another data normalization method scales the data in the range \([0,1]\), referred as cepstral scaling (CS), which can be given as

\[ c_{CS}(i, t) = \left(\frac{c(i, t) - \min(c(i))}{\max(c(i)) - \min(c(i))}\right) \]  

where \( \max(c(i)) \) and \( \min(c(i)) \) are the maximum and minimum data values along the \( i \)th dimension, respectively. These formulas also apply to normalization of delta and delta-delta coefficients.

In [2], the audio is first classified as silent and non-silent where silent is defined as one which has the sum of the signal magnitude below a certain threshold. The mean and standard deviation of the features extracted from the non-silent features are then concatenated to form the feature vector with normalized values. The leave-one-out test is carried out first where each of the 409 sounds is used as query but the query sound is not used as a prototype. A combination of cepstral features and perceptual features, which includes total spectrum power, subband powers, brightness, bandwidth, and pitch, gives an improved performance than the individual features. The combined feature vector gives an error rate of 9.78% which is better than the error rate of 19.07% for the Muscle Fish system [11]. In the second test, evaluation is done using separate training and test sets, 211 files and 198 files, respectively. An error rate of 9.60% is achieved which is once again using a combination of cepstral and perceptual features.

While MFCCs are still probably the most common feature in both speech and sound recognition applications, it has been shown to perform poorly in noisy conditions [9,22]. Multi-conditional training is one solution to this problem but it requires large datasets to capture the variations in environmental conditions.

A commonly used solution seen in speech recognition is root compression. Log spectrum compression used in conventional cepstral analysis is sensitive to noise [23]. The peaks of the mel-filter bank energies are important in characterizing the sound but the log compression can create high variations in the cepstral coefficients for low energy components [24]. Root cepstrum was proposed in [25] to improve the robustness of MFCCs. Root compressed cepstral coefficients are computed similar to the conventional method but root compression is applied to the filter bank energies instead of log compression which can be given as

\[ c(i, t) = \left(\frac{1}{\sqrt{\sum_{m=1}^{M_i} E(m, t)}}\cos\left(\frac{\pi}{M_i}(m - 0.5)\right)\right), \quad i = 1, 2, \ldots, L \]  

where \( \gamma \) is the root value used to compress the filter bank energies, \( 0 < \gamma \leq 1 \). In the event \( \gamma = 1 \), the filter bank energies are uncompressed which is often referred as linear cepstral coefficients. When evaluated on an audio surveillance database in [8], linear MFCCs were shown to give significant improvement in classification accuracy at low signal-to-noise ratios (SNRs) and also found to be more effective for feature vector combination.

In some other variations of MFCCs, in [26], independent component analysis (ICA) MFCCs were proposed for recognizing home environment sounds under air-conditioner noise for home automation. In [27], power normalized cepstral coefficients (PNCCs) were shown to outperform MFCCs under various noise conditions and noise levels, including reverberant environments.

2.2.2. GTCCs

Gammatone cepstral coefficients (GTCCs) are a more recent addition to the family of cepstral features. GTCCs employ a
gammatone filter, a linear filter which models the frequency selectivity property of the human cochlea. The most commonly used cochlea model is that proposed by Patterson et al. [29] which is a series of bandpass filters with the bandwidth given by an equivalent rectangular bandwidth (ERB). An efficient implementation of the gammatone filter bank has been provided in [30] which has been closely followed in speech [31] and non-speech [32] recognition applications.

Extraction of GTCCs follows the same procedure as MFCCs except that gammatone filters are used instead of mel-filters. The impulse response for the gammatone filter can be given as

\[ g(t) = A e^{-\frac{t^2}{2\sigma^2}} \cos(2\pi f_\text{c} t + \phi) \]

where \( A \) is the amplitude, \( f \) is the order of the filter, \( B \) is the bandwidth of the filter, \( f_\text{c} \) is the center frequency of the filter, \( \phi \) is the phase, and \( r \) is the time.

The ERB is used to describe the bandwidth of each cochlea filter in [29]. ERB is a psychoacoustic measure of the auditory filter bank at each point along the cochlea and can be given as

\[ f_{\text{ERB}} = \left( \frac{f_{\text{c},\text{ERB}}}{Q_{\text{ERB}}} \right)^p \left( B_{\text{min}} \right)^p + B_{\text{min}}^p \]

where \( Q_{\text{ERB}} \) is the asymptotic filter quality at high frequencies and \( B_{\text{min}} \) is the minimum bandwidth for low frequency channels. The bandwidth of a filter can then be approximated as \( B = 1.019 \times f_{\text{ERB}} \). The three commonly used ERB filter models are given by Glasberg and Moore [33] \( (Q_{\text{ERB}} = 9.26, B_{\text{min}} = 24.7, \text{and} \ p = 1) \), Lyon’s cochlea model as given in [34], \( (Q_{\text{ERB}} = 8, B_{\text{min}} = 125, \text{and} \ p = 2) \), and Greenwood [35] \( (Q_{\text{ERB}} = 7.23, B_{\text{min}} = 22.85, \text{and} \ p = 1) \).

The cochlea has thousands of hair cells which resonate at their characteristic frequency and at a certain bandwidth. In [30], the mapping between center frequency and cochlea position is determined by integrating the reciprocal of (11) with a step factor parameter to indicate the overlap between filters. This can then be inverted to find the mapping between filter index and center frequency which can be given as

\[ f_{\text{cm}} = -Q_{\text{ERB}} B_{\text{min}} + \left( f_s + Q_{\text{ERB}} B_{\text{min}} \right) e^{-m Q_{\text{ERB}}}, \quad m = 1, 2, ..., M_2. \]

where \( f_s \) is the maximum frequency in the filter bank, \( M_2 \) is the number of gammatone filters, and \( s \) is the step factor given as

\[ s = \frac{Q_{\text{ERB}}}{M_2} \log \left( \frac{f_s + Q_{\text{ERB}} B_{\text{min}}}{f_s + Q_{\text{ERB}} B_{\text{min}}} \right) \]

where \( f_s \) is the minimum frequency in the filter bank.

A 4th order gammatone filter with four filter stages and each stage 2nd order digital filter is described in [30] and an implementation provided in the Auditory Toolbox for Matlab [36].

Valero and Aulas [32] performed a detailed analysis on MFCCs and GTCCs and concluded that GTCCs are more effective than MFCCs in representing the spectral characteristics of non-speech audio signals, especially at low frequencies.

2.3. Time–frequency image features

Every sound signal produces a unique texture which can be visualized using a spectrogram image. The intensity values of the spectrogram image represent the dominant frequency components against time. This can be utilized to improve the recognition rate of sounds in the presence of additive noise, provided the noise spectrum does not contain strong spectral peaks which corrupt the dominant sound signal components. This was demonstrated by Paliwal [37] where spectral subband centroids (SSCs) were used as supplementary features for improved robustness in speech recognition. Some similar works in ASR are discussed below.

2.3.1. Central moments

For sound event recognition, Dennis et al. [9] extract central moments as features from the spectrogram image of sound signals which they refer as the spectrogram image feature (SIF). They consider both grayscale and quantized spectrogram image representations. To obtain the spectrogram image, the linear values are firstly obtained from the DFT values as

\[ S_{\text{linear}}(k, t) = |X(k, t)| \]

\[ S_{\text{log}}(k, t) = \log |X(k, t)|. \]

These values are normalized in the range \([0, 1]\) which gives the grayscale spectrogram image intensity values. The normalization is given as

\[ l(k, t) = \frac{S(k, t) - \min(S)}{\max(S) - \min(S)} \]

These values are then quantized and mapped onto the red, green, and blue (RGB) monochrome components of a color space. The HSV color space was used in this case and the mapping of the grayscale image to the monochrome image can be given as

\[ \eta_q(k, t) = h(l(k, t)) \quad \text{where} \quad q = (1, 2, ..., 8) \]

where \( \eta_q \) is a monochrome image \((R, G, \text{or} B)\), \( h \) is a nonlinear mapping function, and \( q \) the quantization regions.

Each image is then partitioned into \(9 \times 9\) blocks and second and third central moments are computed in each block as features. The \(r\)th central moment for any given block of image can be determined as

\[ \mu_r = \frac{1}{Z} \sum_{i=1}^{Z} (l_i - \mu)^r \]

where \( Z \) is the sample size or the number of pixels in the block, \( l_i \) is the grayscale intensity value of the \(z\)th sample in the block, and \( \mu \) is the mean grayscale intensity value of the block.

For experimentation, 60 sound classes are taken from the Real World Computing Partnership (RWCP) Sound Scene database in Real Acoustic Environment [38] to give a selection of collision, action, and characteristics sounds. Each class has 80 files of which 50 files are randomly selected for training and 30 files are used for testing. Four noise types: speech buzz, destroyer control room, factory floor 1, and jet cockpit 1 from NOISEX-92 database [39] are added at 20 dB, 10 dB, and 0 dB SNRs to test the robustness of the proposed approach. While MFCCs were seen to produce higher classification accuracy under clean conditions, the results using the SIF method were much better at low SNRs. The best results for training with clean signals were between 74% and 77% at 0 dB SNR for the four noise types. This was achieved using the SIF method, quantized linear spectrogram, and HMM classification, implemented using the HTK toolkit [40]. For babble noise, with multi-condition training, training with clean signals and at 20 dB and 10 dB SNRs, the accuracy at 0 dB SNR increased from 74.4% to 79.4% which was using the quantized log spectrogram.

The SIF with reduced feature dimensions, referred as reduced SIF (RSIF), is proposed for an audio surveillance application in [8]. For the RSIF, the mean and standard deviation values of the central moment values are computed along the rows and columns of the blocks. These are concatenated to form the feature vector which is 2.25 times smaller than the SIF but without compromise in classification performance. Experiments are carried out on 10 sound classes: alarms, children voices, construction, dog barking, footsteps, glass breaking, gunshots, horn, machines, and phone rings. Each class contains multiple subclasses and the sound files are largely obtained from the RWCP Sound Scene database in Real Acoustic Environment [38] and the BBC Sound Effects library [41]. The performance is evaluated under three different noise
environments taken from the NOISEX-92 database [39]; speech babble, factory floor 1, and destroyer control room. The feature vector combination of RSIF and linear MFCCs produced significantly better results, especially at low SNRs, with a classification accuracy of 97.11%, 96.06%, 93.61%, 89.15%, and 70.95% under clean conditions and at 20 dB, 10 dB, 5 dB, and 0 dB SNRs, respectively.

2.3.2. GLCM

Gray-level co-occurrence matrix (GLCM), also known as gray-tone spatial dependence matrix [42], is an image processing based texture analysis technique which has been extended to texture analysis of sound signal time–frequency images. GLCM gives the spatial relationship of pixels in an image. It is a matrix of frequencies where each element (i, j) is the number of times intensity value j is located at a certain distance and angle, given by the displacement vector (dx, dy), where dx is the offset in the y direction and dy is the offset in the x direction, from intensity value i in an \( N_x \times N_y \) image \( I \). Mathematically, this can be given as

\[
p(i, j) = \sum_{k=1}^{N_x} \sum_{l=1}^{N_y} \left\{ \begin{array}{ll} 1, & \text{if } k \cdot t \cdot d_x + l \cdot t \cdot d_y = j \\ 0, & \text{otherwise} \end{array} \right.
\]

where the size of the output matrix is \( N_x \times N_y \), and \( N_x \) is the number of quantized gray levels. The typical angles for computing the GLCM are 0°, 45°, 90°, and 135° corresponding to the displacement vector \( [0, d], [-d, d], [-d, 0], \) and \( [d, -d] \), respectively.

After computing the GLCM, the fourteen textural descriptors proposed in [42], a subset of these descriptors, or new descriptors are often extracted as features. This approach has been utilized in a number of applications involving image texture analysis and some recent examples include diagnosis of abdominal tumors using texture classification of ultrasound images [43] and mammogram texture classification for breast cancer detection [44].

Costa et al. [45] applied this technique to texture classification of spectrogram images for music genre recognition. Their audio database consists of 900 music pieces from 10 music genres taken from the Latin music database [46]. The audio signal is first converted to a spectrogram using time decomposition [46] and the GLCM texture descriptors are extracted as features using a zoning technique, that is, the spectrogram image is divided into horizontal sections, with a total of 10 zones, and analysis is carried out in each zone. The following seven textural descriptors are utilized: entropy, correlation, homogeneity, third order momentum, maximum likelihood, contrast, and energy. The results are compared against those in [47] which takes an instance-based approach with feature vectors represented by short-term, low-level characteristics of the music audio signal. Only a marginal increase is seen in the average classification accuracy, increasing from 59.6% to 60.1%, but results showed an improvement of about 7% with a combination of the two methods.

In a face recognition problem [48], however, instead of extracting the textural descriptors as features, the matrix values itself form the feature vector. This was generally shown to give significantly better results than using the combined fourteen textural descriptors as features. This technique was adopted in an audio surveillance application in [49] which was referred as the spectrogram image texture feature (SIFT). Analysis was performed independently in frequency subbands and the final feature vector is a concatenation of the feature vectors from each subband. When tested with the same databases and experimental setup as in [8], the SIFT gave significantly better classification performance at low SNRs and a better overall performance when compared to MFCCs, SIF, and RSIF.

2.3.3. Other time–frequency representations

Short-time Fourier transform (STFT) is probably the most commonly used time–frequency image representation. In this representation, the frequency components are equally spaced with constant bandwidth. However, most sound signals hold greater frequency components in the lower frequency range and less frequency components in the upper frequency range. As such, some information is lost in the lower frequency components while the higher frequency components hold little information in this time-frequency representation.

Mel-spectrogram and gammatone-spectrogram, also known as cochleagram, are variations of the STFT spectrogram utilizing the mel-filter and gammatone filter, respectively. Both these filters have more frequency components in the lower frequency range with smaller bandwidths and fewer frequency components in the higher frequency range with longer bandwidths. This makes their corresponding time-frequency representation more suitable for feature extraction. Cochleagram image-based feature extraction, in particular, has found usage in speech recognition [50] and audio separation [51] applications. This time-frequency representation was also used in an audio surveillance application in [52]. After filtering the signal with the gammatone filter, the energy in the windowed signal for each frequency component is added which can be given as

\[
C(m, t) = \sum_{n=0}^{N-1} \tilde{x}(m, n) w(n) \quad m = 1, 2, ..., M_2
\]

where \( \tilde{x}(n) \) is the gammatone filtered signal and \( C(m, t) \) is the mth harmonic corresponding to the center frequency \( f_{cm} \) for the \( t \)th frame.

These values are then normalized using (16) to get the grayscale cochleagram image intensity values. Cochleagram image feature extraction improved the average classification performance of SIF, RSIF, and STFT, now referred as CIF, RCIF, and CTIF, respectively, from 75.89%, 81.08%, and 85.62% to 86.30%, 89.03%, and 89.24% with significantly improved results at low SNRs. Further improvement in classification performance was also achieved when combined with linear GTCMs. Illustration of spectrogram and cochleagram image for a construction sound signal can be found in Fig. 2 [52].

Wavelet transform [53] also provides time-frequency representation of a signal and has an advantage over Fourier transform in that it provides better time and frequency localization. Nilufar et al. [54] use wavelet packet decomposition [55], an extension of wavelet transform that includes more signal filters, for robust speech and music discrimination. This technique is applied to the spectrogram to transform it into different subbands containing texture information. Multiple kernel learning (MKL) [56] is used to select the optimal subbands for discriminating the two classes.

2.4. Sparse decomposition

Sparse decomposition aims to decompose a given input signal as a linear combination of a defined number of elementary signals from a large linearly dependent collection. While there are a few algorithms for this, such as basic pursuit (BP) [57], matching pursuit (MP) seems to be the most often used in ASR applications. Chu et al. [10] consider MP in environmental sound recognition. Their sound database consists of fourteen environment types, taken from BBC sound effects library [41] and the Freesound project [58], which are as follows: inside restaurants, playground, street with traffic and pedestrians, train passing, inside moving vehicles, inside casinos, street with police car siren, street with ambulance siren, nature-daytime, nature-nighttime, ocean waves, running water/stream/river, raining/shower, and thundering.
In simple terms, MP, originally proposed by Mallat and Zhang [59], allows extraction of time–frequency features through the sparse linear expansion of a waveform. This is done by decomposing signals using an overcomplete dictionary of functions, such as Gabor dictionary [59] as used in their work. Other available dictionaries are wavelets [60], wavelet packets [61], multiscale Gabor dictionaries [62], and chirplets [63]. An overcomplete dictionary ensures that the signal converges to a solution with zero residual energy and, therefore, results in the best set of functions to approximate the original representation.

As explained in [10], given a dictionary \( D \) containing parameterized waveforms \( \phi_x \), as

\[
D = \{ \phi_x : x \in \Gamma \}
\]

(21)

where \( \Gamma \) is the parameter set and \( \phi_x \) is called an atom, the approximate decomposition of a signal \( x \) can then be given as

\[
x = \sum_{i=1}^{a} \alpha_i \phi_{x_i} + R^0, \quad i = 1, 2, ..., a
\]

(22)

where \( R^0 \) is the residual. Given \( x, a, \) and \( D \), the goal is to determine indices \( \gamma_i \) and compute \( \phi_{x_i} \) while minimizing \( R^0 \) with the initial approximation \( x^{00} = 0 \) and \( R^{00} = a \). The sequence of sparse approximation is done stepwise using MP.

In [10], frequency and scale parameters are extracted from each atom as features together with the mean and standard deviation for each parameter, with five determined as the optimum number of atoms. A combination of MFCCs and MP features produced the highest classification accuracy at 83.9% using GMM classification.

Interestingly, they also gave a listening test to 18 individuals who produced an overall accuracy of 77%, 82%, and 85% for an audio clip of 2, 4, and 6 s, respectively. The confidence level of the individuals was also measured with each answer which showed direct correlation with the accuracy. Potential shortfalls in the listening test, such as short duration of clips, were discussed against the results in [64] where listening tests produced better results than the ASR system.

In addition, Scholler and Purwins [65] consider MP for signal approximation with sparse optimization method [66] for drum sound classification. Data samples from ENST database [67] and RWC Music Database: Musical Instrument database [68] are used and the following features are considered: MP features using a sparse coding dictionary (SC-MP), MP feature using a gammatone dictionary (GT-MP), and timbre descriptors (TD). Apart from the three individual feature sets, the combination of TD with SC-MP and GT-MP is also considered. Results are compared under clean conditions and at 20 dB, 10 dB, 0 dB, and –10 dB SNRs with the addition of white Gaussian noise. When trained with clean samples only, the overall performance of the MP features was much better than TD features. While all the features gave comparable results under clean conditions, MP features performed much better under noisy conditions, except at –10 dB and 0 dB SNRs, where all features gave poor results. For TD features, the addition of MP features and multi-conditional training improved the classification accuracy but the overall performance of the individual MP features was still better.

2.5. Feature optimization

The addition of new features to a baseline feature set does not necessarily improve the classification accuracy as some features or feature dimensions are ineffective, depending on the application. Optimization techniques have been used in some literature to determine the optimal feature set. For musical instrument classification, Essid et al. [69] consider various features which can be broadly classified as: temporal, cepstral, spectral, amplitude modulation (AM) [70], and octave band signal intensities (OBSI) [69]. Two feature selection methods, inertia ratio maximization using feature space projection (RMFSP) [71] and genetic algorithms (GAs) [72,73] are experimented with. Class pairwise feature selection with a fusion of the two optimization techniques was found to be most effective in discriminating between possible pair of instrument classes, giving a better overall performance as a result.

Alexandre et al. [5] argue the computational limitations of digital signal processing hardware in a hearing aid application. GA with restricted search [74] is proposed to select the optimal features so that the feature vector dimension could be reduced and the computation speed increased as a result. Three main classes are considered: speech in quiet, speech in noise, and noise. A two-layer structure is adopted for classification. The first layer distinguishes between speech and non-speech (noise) and the second layer classifies speech files into either a quiet environment or a noisy environment. They consider 38 features, mostly cepstral and time and frequency domain features, with the final feature vector 76-dimensional. The results show that while the unconstrained GA required 43 and 46 features to get the best probability of correct classification for the two classification problems, respectively, only 11 features are shown to give comparable performance using restricted GA, which is also always slightly better than the sequential methods [75], sequential forward search (SFS) and sequential backward search (SBS).
3. Classifiers

In this review, we focus on two relatively new classifiers, SVMs and DNNs. We provide a brief background on the two classifiers and discuss their classification performance.

3.1. Support vector machines

3.1.1. Binary SVM

SVM is a statistical learning classifier developed for binary classification. The initial SVM was a linear classifier proposed by Vapnik and Lerner in 1963 [79]. This was extended to nonlinear datasets by Boser, Guyon, and Vapnik in 1992 [80] and has gained widespread attention since the late 90s, around the same time research in ASR was generating interests. We provide a theoretical background of binary SVMs and then discuss the multiclass classification methods of SVMs. We then compare the performance of SVMs against other classification methods and also compare the performance of the multiclass classification methods.

3.1.1.1. Basic theory. As described in [80–82], a support vector machine determines the optimal hyperplane to maximize the distance between any two given classes. Starting with a case of linearly separable dataset, consider a set of training samples belonging to two classes given as \(\{(x_i, y_i)\}_{i=1}^l\), where \(x_i \in \mathbb{R}^d\) is a d-dimensional feature vector representing the \(i\)th training sample, and \(y_i \in \{-1, +1\}\) is the class label of \(x_i\). The optimal hyperplane can be determined by minimizing \(\frac{1}{2} \|w\|^2\) subject to \(y_i(w \cdot x_i + b) \geq 1\), where \(w \in \mathbb{R}^d\) is a normal vector to the hyperplane and \(b\) is a constant. Solving this using classical Lagrangian duality gives the solution

\[
w = \sum_{i=1}^l \alpha_i y_i x_i
\]

where \(\alpha_i\) are the non-negative Lagrange multipliers. The \(x_i\) for which \(\alpha_i > 0\) are the support vectors which lie exactly on the margin satisfying \(y_i(w \cdot x_i + b) = 1\), \(i = 1, 2, \ldots, l\). The offset can then be determined as

\[
b = y_i - w \cdot x_i
\]

using any support vector or averaged over all support vectors.

For linearly nonseparable problems, the optimization can be generalized by introducing the concept of soft margin [81]. Introducing non-negative slack variables \(\xi_i\) which measure the degree of misclassification of data \(x_i\) and a penalty function \(\sum_i \xi_i\), the optimization is a trade-off between a large margin and a small error penalty. The solution is similar to the separable case except for a modification to the Lagrange multipliers:

\[
0 \leq \alpha_i \leq C_i, i = 1, 2, \ldots, l,
\]

where \(C_i\) is a penalty or tuning parameter to balance the margin and training error.

In applications where linear SVM does not give satisfactory results, nonlinear SVM is suggested which aims to map the input vector \(x\) to a higher dimensional space \(z\) through some nonlinear mapping \(\Phi(x)\) chosen a priori to construct an optimal hyperplane. The kernel trick [80] is applied to create the nonlinear classifier where the dot product is replaced by a nonlinear kernel function \(K(x_i, x_j)\) which computes the inner product of the vectors \(\Phi(x_i)\) and \(\Phi(x_j)\). A commonly used kernel function is Gaussian radial basis function (RBF) given as \(K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})\), where \(\sigma > 0\) is the width of the Gaussian function.

The classifier for a given kernel function with the optimal separating hyperplane is then given as

\[
f(x) = \text{sgn}\left(\sum_{i=1}^l \alpha_i y_i K(x_i, x) + b\right)
\]

3.1.1.2. Multiclass classification. Being a binary classifier, a number of techniques have been proposed for multiclass classification. The most common technique is to reduce the multiclass classification problem into multiple binary classification problems. Four commonly used methods based on this technique are one-against-all (OAA), one-against-one (OAO), decision directed acyclic graph (DDAG), and adaptive directed acyclic graph (ADAG).

OAA is probably the earliest of the multiclass SVM classification techniques [82]. For a P-class problem, P binary SVM classifiers are constructed and evaluated where the \(i\)th classifier is trained with all the training samples from the \(i\)th class as positive labels and all the remaining samples as negatives labels. During classification, a sample \(x\) is classified in the class with the largest value of the decision function which can be given as

\[
f(x) = \arg\max_{i=1,\ldots,P} \sum_{j \neq i} \alpha_j y_j K(x_j, x) + b_j.
\]

The OAO approach distinguishes between every pair of classes and classification is done using the max-wins voting strategy [83]. For a P-class problem, OAO-SVM constructs and evaluates \(P(P-1)/2\) classifiers where each SVM is trained on samples from two classes at a time, that is, using training samples from the \(i\)th and \(j\)th classes. During classification, the class label of a test sample is predicted as

\[
f(x) = \arg\max_{i=1,\ldots,P} \sum_{j \neq i} \text{sgn}\left(w^j \cdot \Phi(x) + b^j\right)
\]

DDAG [84] and ADAG [85] are also based on classification between pair of classes but utilize a decision tree structure in the testing phase. Similar to OAO-SVM, \(P(P-1)/2\) nodes are created during training phase but only \(P-1\) nodes are used during evaluation.

3.1.1.3. Comparison of SVM with other classifiers. Guo and Li [3] use the Muscle Fish database and similar features as in [2], that is, cepstral and perceptual features. However, a new metric called distance-from-boundary (DFB) is proposed for audio retrieval using SVMs to learn the boundaries. Using the same feature vector formation technique as [2], SVM performed better than NN, kNN, and NC (nearest center) classifiers. The lowest error rate is 11.00% for the leave-one-out test but 8.08% with separate training and test sets.

In another similar work, Lu et al. [86] consider five audio classes: silence, music, background sound, pure speech, and non-pure speech. SVM, with a Gaussian RBF kernel, is used for classification. For experimentation, a database with 2600 audio clips is created with a total duration of about 4 hours obtained from TV programs,
In general, SVM has been shown to give a better classification performance than most traditional classifiers. In [9], HMM gives a better classification performance than SVM, however, they used linear SVM. Linear SVM is generally considered not suitable for complex classification tasks and most other similar works use nonlinear SVM. Results in [17] show that there is not a significant difference between HMM and binary SVMs. In addition, it is important to tune the parameters $C$ and $\sigma$ correctly, avoiding local minima, for best results.

3.1.1.4. Comparison of multiclass classification methods. There are various pattern recognition problems where multiclass SVM classification methods have been compared. Hsu and Lin [89] compare the performance of OAA, OAO, DDAG and two altogether methods, an approach for multiclass problems by solving a single optimization problem, on large classification problems by solving a single optimization problem, and evaluation time when compared to OAA and R-OAA. An optimized DAG method is proposed by Takahashi and Abe [96]. Another improvement to DAG, based on binary decision tree and Huffman code [97], is given by Chen and Liu [98]. Fei and Liu [99] propose a binary tree of support vector machine (SVM) which uses one-to-one training scheme but reduces the total number of classifiers by employing a probabilistic model. It gives a higher classification efficiency, that is, it gives comparable classification accuracy to the traditional methods but is generally faster.

In general, the difference in the classification accuracy of the various multiclass SVM classification methods which have been proposed is marginal and the preference of one method over the others is largely based on faster training and/or evaluation times. However, most of these comparisons are limited to clean conditions only. In [8], the performance of OAA, OAO, DDAG, and ADAG multiclass SVM classification methods are compared under noisy conditions. The OAA classification method is seen to be the most noise robust and also gave a significantly better performance with feature vector combination. In addition, it required a significantly lower evaluation time than OAO but was slower than DDAG and ADAG methods.

3.1.1.5. Hybrid classifier. Use of a hybrid SVM/kNN classifier using MPEG-7 audio descriptors can be found in [100]. MPEG-7, formally called “Multimedia Content Description Interface”, is a multimedia content description standard [101] which provides a comprehensive range of descriptors and descriptor schemes ranging from low level audio and video features to high level semantic features.

Three MPEG-7 audio low-level descriptors, spectrum centroid, spectrum spread, and spectrum flatness, are used as features for classifying 12 sound classes: male speech, female speech, cough, laughing, screaming, dog barking, cat meowing, frog wailing, piano, glass breaking, gun shooting, and knocking. A frame-based classification strategy is used for the hybrid classifier where the output of the SVM classifier, using a RBF kernel and DDAG multiclass classification method, and kNN classifier are turned into probabilistic scores which are used to get a combined frame score. An unknown sound signal is then assigned to the class which most of the frames are assigned to. With 50% of the samples used for testing and 50% for testing, a maximum classification accuracy of 85.1% is achieved using the hybrid classifier when compared to a maximum of 83.2% using HMM classification. However, the results are not compared to the standard SVM and kNN classification methods.

### 3.1.2. One-class SVM

Unlike the binary SVMs considered so far, one-class SVM (1-SVM) is used in an audio surveillance application by Rabaoui et al. [17]. 1-SVM, proposed by Schölkopf et al. [102], is a modification of binary SVM to solve one-class classification problem. The feature is transformed by the kernel and the origin is treated as the second class. 1-SVM essentially separates the feature data points from the origin and maximizes the distance from this hyperplane to the origin.

1-SVM is more suited with high dimensional feature vectors. As such, unlike most other works where mean and standard deviation values of the extracted features across all frames are concatenated to form the feature vector, a slightly different approach to feature data representation is taken in [17]. The overall feature data for the sound signal is divided into three portions: 30%, 40%, and 30% of the total number of frames. Mean value of the data across each dimension from each portion are concatenated to form the feature vector which results in a feature dimension which is 1.5 times longer than the conventional technique. The classification accuracy of 1-SVM was generally higher than HMM, OAA-SVM, and OAO-SVM classification methods when tested at various SNRs with a number of individual and combined features.

### 3.2. Deep neural networks

While SVMs have seen an increased usage in ASR systems, a new machine learning algorithm called deep learning is gaining a lot of interest in speech recognition. Deep learning aims to learn high-level representations of data through a hierarchy of intermediate representations, such as deep neural networks (DNN) [103]. It has been used for acoustic modeling by research groups at University of Toronto, Microsoft Research, Google, and IBM Research, amongst others, and shown to outperform most other classification methods [104]. Furthermore, in two recent works in acoustics event classification [105,106], DNN was shown to perform better than other classifiers. However, these works compare the classification performance using mel-frequency cepstral coefficients (MFCCs) only.

In another recent work in SER [107], the classification performance of SVM and DNN classifiers are compared with a number of features. These include MFCCs, features extracted from a stabilized auditory image (SAI) [108], and the SIF [9]. The DNN classifier utilizes $L$-layers with the feature vectors on the input layer and output layer in a one-of-$P$ configuration ($P$-classes). The DNN is constructed using individual pre-trained restricted Boltzmann machine (RBM) pairs. Each pair comprises V visible and H hidden neurons.
stochastic nodes, $v = [v_1, v_2, \ldots, v_L]^T$ and $h = [h_1, h_2, \ldots, h_L]^T$. Bernoulli-Bernoulli RBM structures are used on the intermediate and final layers while the input layer is Gaussian-Bernoulli RBM. Given the energy functions of the two RBM structures, $E(v, h)$, the joint probability associated with configuration $(v, h)$ is given as

$$p(v, h; \theta) = \frac{1}{Y} e^{-E(v, h; \theta)}$$

where $Y$ is a partition function given as $Y = \sum \sum e^{-E(v, h; \theta)}$.

The training data is used to estimate the RBM model parameters $\theta$ by maximum likelihood learning using the contrastive divergence (CD) algorithm [103]. After pre-training, a size $P$ softmax output labeling layer is added to the pre-trained stack of RBMs [109] to convert a number of Bernoulli distributed units in the final layer into multinomial distribution. The stacked network is then trained using back propagation.

DNNs generally gave significantly higher overall classification accuracy than SVM with the best overall feature performance achieved using the SIF. With multi-condition training, for example, the average classification accuracy using SVM is 88.55% but 92.58% with DNN. Similar conclusions were also drawn with MFCCs and SAI features.

4. Discussion and conclusion

In this section, we first provide a summary of some of the key advancements in ASR. We then discuss some relatively lesser known applications of ASR.

4.1. Summary of advancements

MFCCs have evolved as a baseline feature in many ASR systems. However, it is often supplemented with other features such as perceptual features [2,3] and MP-based features [110] for improved classification performance. A combination of linear cepstral coefficients and time-frequency image features has also shown to give robust performance in [8,52]. While SVMs have been the preferred classifier in most ASR applications, DNNs have gained popularity in recent years with its superior classification performance, as demonstrated in a number of pattern recognition problems. In Table 1, we summarize what we believe are some of the key works in ASR and highlight the advancements in features and classifiers using the work by Wold et al. [1] as basis.

While there has been a decent amount of development in ASR systems over these years, the inconsistency in the choice of sound databases in most literature makes it difficult to make direct comparison of the performance of the proposed techniques. While

Table 1: An overview of some key works in ASR.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Year</th>
<th>Application</th>
<th>Sound Database(s)</th>
<th>Noise Database(s)</th>
<th>No. of Classes</th>
<th>Total Files</th>
<th>% Training data</th>
<th>Best feature(s)</th>
<th>Classifier</th>
<th>Accuracy (or Error Rate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wold et al.</td>
<td>1996</td>
<td>Content-based audio classification</td>
<td>Muscle Fish</td>
<td>–</td>
<td>16</td>
<td>409</td>
<td>–</td>
<td>Perceptual features</td>
<td>NN</td>
<td>19.07% (ER)*</td>
</tr>
<tr>
<td>Li [2]</td>
<td>2000</td>
<td>Content-based audio classification</td>
<td>Muscle Fish</td>
<td>–</td>
<td>16</td>
<td>409</td>
<td>Leave-one-out test</td>
<td>MFCC + perceptual features</td>
<td>NF</td>
<td>9.78% (ER)</td>
</tr>
<tr>
<td>Guo and Li [3]</td>
<td>2003</td>
<td>Content-based audio classification</td>
<td>Muscle Fish</td>
<td>–</td>
<td>16</td>
<td>409</td>
<td>Leave-one-out test</td>
<td>MFCC + perceptual features</td>
<td>SVM</td>
<td>11.00% (ER)</td>
</tr>
</tbody>
</table>

* ER as reported in [2,3].

** Different combination of features are experimented with under clean and noisy conditions.

*** Average classification accuracy value over four noise types.
sound libraries, such as the Latin music database, RWCP databases, and the BBC sound effects library have been employed for research in certain ASR applications, the creation of the sound database for use from these available libraries is at the discretion of the researchers. Also, the number and complexity of sound classes and the amount of training data, amongst others, have a direct influence on the classification performance of an ASR system. Therefore, there is a need to standardize sound databases and experimental setups to make it easier for direct comparison of proposed techniques, similar to what was seen in [1–3], refer to Table 1.

In addition, different approaches have been noticed in structuring of classes in some similar applications, such as audio surveillance and sound event recognition. For an audio surveillance application in [8,17], a sound class has a number of sound events. For example, shots fired from a rifle, shotgun, and machine gun are examples of different sound events but would be treated as a single sound class such as gunshots. In some cases, the signal properties of subclasses in a particular class are similar to the subclasses in other classes but different to subclasses in its own class. This creates interclass similarity and intraclass diversity, increasing the complexity of the problem as a result.

Moreover, there has been a lack of attention in recognition of overlapping sound events. An example of such a system is presented in [111] and, while there are some other similar works, more research is needed in this area.

4.2. Other applications of ASR

The applications of ASR are not limited to content-based audio classification, audio surveillance, sound event recognition, and environmental sound recognition, which have been the focus of our review so far. A summary of some less conventional applications of ASR is given in Table 2.

Similar to fingerprint recognition, face recognition, and, more recently, vein pattern recognition systems, heart sound recognition has the potential for human identification. Heart and lung sound recognition can also be used for diagnosis of disorders associated with the heart and lung, respectively. This extends to animals as well such as for identification of respiratory infections in pigs and dairy calves. Such technology can act as an early warning system which could help contain contagious viruses with some viruses from animals, such as swine flu, known to affect humans as well. Diagnosis of disorders using ASR technology extends beyond heart and lung sound recognition. An example of gastrointestinal motility monitoring system using bowel sounds,
captured through abdominal surface vibrations, can be found in [147].

In addition, ASR can be used for animal species recognition through analysis of their call sound. Such a system can be used to carry out automatic animal species monitoring replacing the laborious manual recognition process. It could also be used for environmental monitoring since the abundance of wildlife generally indicates a healthy environment and vice-versa. For example, researchers in Brisbane established a sensor network in the city’s suburbs and forest park to study the impact of urbanization of neighboring suburbs on the ecological system, with the focus on recognizing bird species using acoustic signals [111].

Furthermore, while we have generally been looking at examples of standalone ASR systems so far, audio and video recognition systems could also be integrated for a more holistic approach to this problem, such as in the development of surveillance systems. Video surveillance systems have been around for many years but have limitations such as limited field of view and relatively expensive computation and data storage. An ASR system could be used to complement a video-based recognition system such as in public transports [139], detecting traffic events [140], fall detection [141], machine awareness [142], and in banks [143]. Audio and video recognition systems could also be combined for recognition of complex events in real movies [144]. Another such example is robotics. Robots are often aimed at mimicking human behavior and, similar to humans, acoustical information can be utilized to make a more complete description of the scene. Robotics based search and rescue operation is one such scenario where in the aftermath of a natural disaster, such as an earthquake, the injured could be hidden collapsed structures and audio information such as screaming or crying could be used to reach them [145].

References

Fast matching pursuit with a multiscale dictionary of Gaussian
P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, D. Martinez-Munoz,
S. Arivazhagan, L. Ganesan, Texture classification using two-dimensional non-negative matrix factorization with automatic
R.V. Sharan, T.J. Moir, Subband time-frequency image texture features for
2605–2615.
S. Elad, N. Ray, M.K.J. Molla, and K. Hirose, Spectrogram based features
selection using multiple kernel learning for speech/music discrimination,
A. Arvazhagan, L. Ganesan, Texture classification using wavelet transform
the kernel matrix with semidefinite programming, J. Mach. Learn. Res. 5
S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit,
M. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries,
P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, D. Martínez-Munoz, F. Lopez-Ferreras, Transient modeling by matching pursuits with a wavelet
dictionary for parametric audio coding, IEEE Signal Process. Lett. 11 (3)
G. Yang, Q. Zhang, P.-W. Que, Matching-pursuit-based adaptive wavelet-packet
atomic decomposition applied in ultrasonic inspection, J. Nondestruct.
R. Gribonval, Fast matching pursuit with a multiscale Gaussian
S. Gholami, D.C. McLeanor, and A. Ayatollahi, Comparing Gaussian and
chirplet dictionaries for time-frequency analysis using matching pursuit
decomposition, in: Proceedings of the 3rd International Symposium on
S. Scholler, H. Purwins, Sparse approximations for drum sound classification,
E. Smith, M.S. Lewicki, Efficient coding of time-relative structure using
G. Gillet and G. Richard, ENST-Drums: An extensive audio-visual database for
drum signals processing, in: Proceedings of 7th International Conference
M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, RWC music database:
Music genre database and musical instrument sound database, in: Pro-
duction of the Fifth International Conference on Natural Com-
putation, 1999.
S. Esfand, G. Richard, D. David, Musical instrument recognition by pairwise
A. Eronen, Automatic Musical Instrument Recognition (Musi:s) thesis, Uni-
G. Peeters, Automatic Classification of Large Musical Instrument Databases
using Hierarchical Classifiers with Inertia Ratio Maximization, 115, Audio
J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory
Analysis with Applications to Biology, Control, and Artificial Intelligence, MI:
J. Fujinaga, Machine recognition of timbre using steady-state tone of acoustic
musical instrument soundings of the International Computer Music
S. Salcedo-Sanz, C. Camps-Valls, F. Perez-Cruz, J. Sepulveda-Sanchis, C.
Bousono-Calzon, Enhancing genetic feature selection through restricted
C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford,
1995.
D.H. Milone, J.R. Galli, C.A. Cangiano, H.L. Ruda
F.V. Gessel. Top 40 Bird Songs [Online]. Available:
J. Cai, D. Ee, B. Pham, P. Roe, and J. Zhang, Sensor network for the monitoring
R. Mankin. Sound Library. Available:
D.F. Specht, Probabilistic neural networks, Neural Netw. 3 (1990) 109
H. Jaafar, D.A. Ramli, Automatic syllables segmentation for frog identi

Tom J Moir was born in Dundee Scotland. He was sponsored by CEC Industrial Controls Ltd, Rugby War
wickshire UK from 1976 to 1979 during his BSc in control engineering which he was awarded in 1979. In 1983 he received the degree of Ph.D for work on self-tuning filters and controllers. From 1982 to 1983 he was with the Industrial Control Unit University of Strathclyde, Scotland. From 1983 to 1999 he was a lecturer then senior lecturer at Paisley College/University of Technology (AUT), New Zealand, under an AUT doctoral scholarship. His research interests include pattern recognition, image processing, and signal processing. He is a member of IEEE.

Ronel V Sharan received the B.Tech. and M.Sc. degrees in electrical and electronic engineering from the University of the South Pacific (USP), Fiji, in 2004 and 2007, respectively. From 2004 to 2005, he was a graduate assistant at the School of Engineering at USP and, as assistant lecturer from 2006 to 2013. He is currently pursuing Ph.D. degree at the Auckland University of Technology (AUT), New Zealand, under an AUT doctoral scholarship. His research interests include pattern recognition, image processing, and signal processing. He is a member of IEEE.