Single-signal entity approach for sung word recognition with artificial neural network and time–frequency audio features

Peerapol Khunarsa

Faculty of Science and Technology, Uttaradit Rajabhat University, Uttaradit, Thailand
E-mail: peerapol@uru.ac.th

Published in *The Journal of Engineering*; Received on 26th May 2017; Accepted on 24th July 2017

Abstract: Singing voice recognition is very different from speech recognition or automatic speech recognition because there are distinct differences between speaking and singing voices. The problem is complex because music audio signals with their background instrumental accompaniments are regarded as noise sources that degrade the performance of the recognition system. This study proposes a statistical learning method to recognise words in a vocal audio signal with background music and to classify the region of a singing voice in a polyphonic audio signal. The goal of this study is to solve the problem of recognising words from sung input without using any method to separate instrumental from the background. This study also applies a concept from image recognition by using a spectrogram feature as an image to solve the problem. An audio signal with accompanying music was analysed and transformed into a spectrogram feature. To recognise it, the entire spectrogram feature was sliced, forming a feature vector for a feed-forward neural network classifier. Several classification functions were compared, including *K*-Nearest Neighbour, Fisher Linear Classifier, Linear Bayes Normal Classifier, Naive Bayes Classifier, Parzen Classifier and Decision Tree. The results show that using a feed-forward neural network can effectively recognise sung words at an accuracy rate of more than 93.0%. In particular, this system can recognise cross-language music data.

1 Introduction

Sung word recognition is one of the interesting research topics in the field of Music Information Retrieval (MIR). The first approach to solve this problem used techniques from automatic speech recognition (ASR). In this paper, we propose a novel technique to solve the problem of singing voice recognition in polyphonic recordings. Our assumption is that it is unnecessary to filter the instrumental background from the singing voice to recognise the words being sung. By taking this approach, we expect to achieve high recognition accuracy.

Singing is the act of producing musically relevant sounds with the human voice. Singing is an augmented version of regular speech because it uses sustained tonality, rhythm, and a variety of other human vocal techniques. The problems involved in recognising words being sung under noisy background conditions, has been a topic of interest to many researchers [1–8] especially the task of recognising words mixed with several musical instruments. Another issue in singing voice recognition is that the problem is quite different from speech recognition (SR) or ASR because of substantial differences between speaking and singing voices such as the duration of vocal sounds, the volume, pitch, vibrato, formant, rhythm and rhyme [9–16]. To make the problem realistic and feasible, we considered singing voices in a polyphonic audio signal sampled from commercial compact-discs (CD) or DVDs of popular music recordings. The sampled set includes a comprehensive list of the commercially pertinent genres in popular music, including dance, soft rock, hard rock, rock, soul, hiphop, R&B, folk and acoustic. All the music samples include a male or female singer and the songs in this study include both Thai and English songs. Another problem in the study of sung word recognition is that a few English and Thai words have special characteristics due to their tone patterns. Different tones have different meanings. For example, depending on the rhythm, the sound of a Thai word may be changed during singing.

The basis of this research involves sound classification techniques. Several techniques have been proposed to solve the problem of sound classification [17–25]. Most of these sound classification methods consist of two processing steps: feature extraction and classification. In the first step, feature exaction, the redundant information in the signal is transformed into descriptors used as the input of a classifier for recognition in the second step. Shenoy [26] used amplitude variation over time in each sub-band and a threshold method on the energy functional the proportion of frames classified as vocals–to predict the proportion of the singing in the entire song. Nwe *et al*. [27] used harmonic attenuated log frequency power coefficients (LFPCs) with hidden Markov models (HMMs) based on three parameters, e.g., section type (intro, verse, chorus, bridge and outro), tempo and volume. Tsai *et al*. [28] used Mel-frequency cepstral coefficients (MFCCs) and GMM models to differentiate vocal from non-vocal signals. Berenzweig and Ellis [29] used the vector of posterior probability as a feature and an HMM framework with two states, ‘singing’ and ‘not singing’. Chou and Gu [30] used 4 Hz modulation energy, harmonic coefficients, 4 Hz harmonic coefficients, delta MFCC and delta log energy as features and used a GMM model to detect the singing voice. Berenzweig *et al*. [31] applied 13 perceptual linear prediction coefficients (PLPCs) and MLP. Maddage *et al*. [32] considered Linear Predictive Coding (LPC), LPC derived cepstra (LPCC), MFCC, spectral power, the short time energy function, and zero-crossing rate (ZCR) as features and a multi-layer neural network consisting of an SVM and a GMM for classification. The SVM was found to outperform the other classifiers. Maddage *et al*. [33] later applied a TISFT (Twice Iterated Composite Fourier Transform to each audio frame. Rocamora and Herrera [34] used different paired sets of features such as MFCCs with their deltas, LFPC with their deltas and double deltas, PLPCs with their deltas, and HC and pitch and tested a variety of classifiers such as an SVM, a back propagation NN, a decision tree classifier, and two different *K*-Nearest Neighbour (KNN) algorithms. Tzanetakis [35] used spectral shape features, MFCCs, mean and deviation of pitch, centroid and LPCs for feature extraction and a naÃ´rve Bayes network, nearest neighbour algorithms, a back-propagation ANN (artificial neural network), a decision tree classifier based on the C4.5 algorithm and SVM classifiers. Kim [36] used a harmonic measure, defined as the ratio of the total signal energy to the maximally harmonically attenuated signal and a threshold method on the harmonic measure to classify segments.

Compared to other research areas in sound classification such as speech, to the best of our knowledge, only a few frameworks have

doi: 10.1049/joe.2017.0210

been proposed to investigate singing voice recognition with background instrumental accompaniment. Most of the prior investigations of singing voice recognition address recognising phonemes first and then use a speech recogniser for lyrics recognition. Sasou *et al*. [10] tested an auto regressive HMM with pure singing voice signals from the RWC database. These studies presumed that the sound involved only pure monophonic singing voices without accompaniment; they did not consider the additional difficulties for practicable use with musical audio signals such as CD recordings. Suzuki *et al*. [37] combined both the melody and the lyrics of a user's singing voice to retrieve a song from a database. They also used a large vocabulary SR system with an HMM as the acoustic model adapted to the singing voice using adaptive speaker technology.

Wong *et al*. [38] proposed a system for real-time alignment of music sung in Cantonese, which is a particular tone language in which the meaning of a word changes when it is pronounced with a different pitch. An MLP was used to segregate vocal from non-vocal segments, using the spectral flux, the HC, the ZCR, the MFCCs, the amplitude level and the 4 Hz modulation energy as input. The DTW algorithm was used to align the two sequences. However, this method is not consistently effective because the durations of uttered phonemes depend on their locations in the music; therefore, they differ even when the phonemes are the same. Kan *et al*. [39] proposed what was probably the first English lyrics sentence level alignment system for aligning lyrics to musical signals for a specific song structure. Gruhne *et al*. [40] proposed a system that performed automatic classification of 15 voiced sung phonemes in polyphonic audio. Their procedure was based on extracting harmonics and re-synthesising a number of partials as a preprocessing step to reduce the influences from the accompanying musical sounds. Then, low-level features were extracted from the audio and classified using classification techniques such as SVM, GMM and MLP. Fujihara *et al*. [41] proposed automatic synchronisation between lyrics and polyphonic music signals for Japanese CD recordings. Their proposed system included detection of vocal segments, segregation of vocals and adapting a speech recogniser to the segregated vocal signals. During the first step, the harmonics were extracted and re-synthesis was performed as in Gruhne *et al*. [40]. A simple HMM was used to preserve only the vocal regions while removing the non-vocal sections. Finally, features extracted from the audio included MFCCs, delta MFCCs and delta power. The Viterbi algorithm was used to align the segmented vocal parts with the corresponding lyrics. Zwan *et al*. [42] applied a neural network and rough sets to solve the problem of an automatic singing voice recognition. However, this approach is computationally complex because the method required and combined many types of feature vectors for classification. Mesaros and Virtanen [43] studied the use of *n*-gram language models to recognise phonemes and words in monophonic and polyphonic music. They considered uni-, bi- and tri-gram language models for phonemes and bi- and tri-grams for words. During the recognition process, an HMM-based phonetic recogniser was adapted to the singing voice. However, their word recognition system achieved a correct recognition rate of only 24%.

The difficulty in recognising lyrics lies with the types of instruments and their power ratio (dB). Therefore, it is possible that the background instrumental accompaniment has been regarded as a noise source that degrades the performance of the recognition system. During a singing period, the power ratio of a singing voice may be stronger or weaker than the power ratio of the music instruments. If the singing voice is stronger than the musical background, the recognition is rather simple. In contrast, it becomes quite complex when the power of the singing voice is weak in relation to the background. Consequently, many methods based on features extracted directly from the accompanied vocal segments have difficulties achieving good performance when the accompaniment is stronger or the singing voice is weaker. To solve these challenging background instrumental accompaniment problems, many studies have used filtering processes or separation tasks to separate the singing voice from the music accompaniment in monaural recordings. The singing voice separation task analyses competing entries to blindly separate the singers voice from music recordings. Several techniques have been proposed that may be relevant to the problem of recognising sung words with a complex musical background. Different researchers have developed several algorithms for separating a voice from musical noise, as summarised below. Many of the existing algorithms utilise the harmonic structure of the singing voice to differentiate the singing pitch from the input mixture for separation purposes. For example, Hu and Liu [44] exploited CASA (Computational Auditory Scene Analysis) to segregate singing voice units for each time frame. Raj [45] applied PLCD (Probabilistic Latent Component Decomposition) to separate singing voices from background music in popular songs. Huang *et al*. [46] proposed RPCA (Robust Principal Component Analysis) to separate singing voices from music accompaniment.

However, the most popular technique for separating singing voices from background music is Non-negative matrix factorisation (NMF) [47, 48]. NMF has often been used to separate a polyphonic spectrogram into non-negative components and then cluster those components into vocal components and accompaniment components. Durrieu *et al*. [49] represented the leading voice using a filter model, while an unconstrained NMF model was used to represent the background music. Imekli and Cemgil [50] presented a Tensor 3 factorisation model for musical source separation. The approach is an extension of NMF in which more than one matrix or tensor object are factorised simultaneously. Their models included spectral information using isolated note recordings or incorporated harmonic information. Following Mohammadiha *et al*. [51], Smaragdis and Leijon used a speech enhancement method based on a Bayesian formulation of non-negative matrix factorisation. They used an HMM in combination with Bayesian non-negative matrix factorisation to derive the MMSE (minimum mean square error) estimator with no information about noise. Then, they learned the BNMF model online and used it to develop an unsupervised speech enhancement system. Many music pieces have repeated musical backgrounds over which varying vocals are superimposed. Coincidentally, Yoo *et al*. [52] and Kim *et al*. [53] applied NMPCF (non-negative matrix partial co-factorisation) to separate drum sources from monaural mixtures of polyphonic music containing various pitched instruments as well as drums. Additionally, other techniques have been proposed for separating polyphonic music such as HPSS (Harmonic-Percussive Sound Separation) [54] and vocal F_0 estimation [55, 56].

This paper focuses on singing voice recognition in polyphonic recordings of popular music. Our hypothesis is that it is unnecessary to filter the instrumental background from the singing voice to recognise the sung words. Because background filtering processes involve high computational costs, the computational complexity in previous works was too high. Because the complexity of musical backgrounds in terms of the relevant factors previously mentioned is too high and uncontrollable, it would be better to find an approach that did not involve eliminating the musical background from the singing voice. Our objectives are concerned with two essential issues. The first issue is the recognition speed. By avoiding having to filter the musical background from the singing voice, we expect the processing time to be greatly reduced. The second issue emphasises the independence of the following factors: the durations of voice sounds, volume, pitch, vibrato, formant, rhythm and rhyme. These two issues lead to the problem of determining which representation domain is the most suitable for any song so that the highest recognition accuracy of the sung words can be obtained. In our algorithm, we transformed the problem of recognising one-dimensional song signals into a problem that involves recognising a colour image. Then, the features of the image are extracted and classified. The details are

discussed in the following sections. The rest of this paper is organised as follows. Section 2 formulates our studied problem and constraints. Section 3 discusses the concepts underlying our proposed algorithm. Section 4 explains the data collection process. Section 5 explains the experimental setup. Section 6 evaluates the results, and Section 7 concludes the paper.

2 Problem formulation and constraints

We considered the following situation. Given a song as a mixture of musical background and singing voice recognise the lyrics. There are two procedures involved in this situation. The first procedure concerns the problem of identifying the duration of each sung word in each song, which can differ depending on the singer and tempo. Then, there is the problem of how to make the duration of each sung word the same. The second problem is how to recognise words in music sung over an instrumental background music with instrumental interference. In polyphonic musical recordings, the instrumental interference is treated as a noise source that degrades the intelligibility of the singing voice signal. The solutions to these two problems are independent from each other. In this paper, we concentrate on both procedures. Hence, it is assumed that the input to our algorithm is an audio signal that already contains a sung word. The input is in the form of a set of sampled audio signal values in a time series, i.e., $\{x\ (1),\ldots,\ x(n)\}$. Our study is constrained by the following factors and conditions.

2.1 Constraints

The problem considered in this paper is defined as follows. Given a song consisting of human singing mixed with instrumental background, detect the points that include the human voice and recognise the word sung at that point.

- Our system took a polyphonic music audio signal as the input sampled from CD music recordings.
- The experiments include different musical genres such as rock, hard rock, soft rock, dance, hip-hop, R&B, soul, folk and acoustic from various artists.
- All the musical genres included either male or female singers.
- The lyrics could be sung in either Thai or English. However, the number of Thai words is large; it is impossible to develop an efficient algorithm to recognise all the possible words. Therefore, only frequently occurring and common words, phases and sentences in most of the sampled songs were considered. Table 2 summarises the frequently used Thai and English words, phrases and sentences and their durations.

2.2 Problems discussed

The problems discussed in this paper are the following. Let $\boldsymbol{S}=\{x(1),\ldots,x(n)\}$ be a given series of sampled signals of a song. Each $x(i)$ may be a mixture of a singing voice with musical background or a singing voice alone.

- Recognise the sung word at point S without eliminating the musical background.
- Find the essential features so that the recognition rate achieves high accuracy.
- Determine whether the recognising algorithm is robust to the previously mentioned constraints.

3 Proposed concept

Recognising a sung word is more complex than recognising a spoken word without any musical background. The strength and clarity of a sung word are always detrimentally affected by several factors such as singing style, the duration of the singing voice and the instrumental background signal which has uncontrollable volume, pitch, vibrato, formant and rhythmic variations. To effectively eliminate the musical background, the types of musical instruments must be known in advance to properly filter the corresponding musical signal frequencies from the signing word signal. However, these frequencies are often unknown prior to the filtering process. If the musical background cannot be completely separated from the singing signal, then the recognition accuracy percentage will obviously not be high. Furthermore, the unpredictability of the singing duration can make the recognition process complicated in terms of time complexity.

Our solution is based on the following observation and hypothesis. The hypothesis is that for a sung word, there are various ways to sing the word with different backgrounds. However, if we plot the spectrograms of all different intervals of songs containing this word and use those as a feature, then the spectrograms should form similar features. Fig. 1 shows some examples of the spectrogram features of the same words. There are four words, named A, B, C and D, and the spectrograms features are illustrated in rows 1–4, respectively. These four words were sung by different performers with different musical backgrounds and durations. However, it is easy to observe that the spectrogram features of any given sung word are similar to each other but different from the spectrogram features of the other sung words. Note that each spectrogram feature was derived from a mixture of a sung word and a musical background. Therefore, it is unnecessary to filter the background from the sung word prior to the recognition process. The spectrogram feature can be considered as a colour image. Using our approach, the problem of recognising sung word with musical background is transformed into the problem of recognising a spectrogram feature. Our recognition algorithm consists of the following steps.

- Modify the time-scales of the input audio signals S to equalise the length of different audio signals.
- Transform the input audio signals S into a spectrogram feature.
- Extract the features used to represent the spectrogram.
- Classify the features.

The results from our proposed technique will be compared with an ASR algorithm. The details of each step are provided in the following section.

3.1 Spectrogram feature representation

In this paper, we applied the concept of recognising audio using a spectrogram feature. A spectrogram feature is a visual representation of the distribution of acoustic energy across frequencies in a time domain. The horizontal axis of a spectrogram feature typically represents the time intervals of audio signal snapshots, while the vertical axis represents the power spectrum of discrete frequency steps. The strength of the power detected is represented as the intensity at each time–frequency pixel.

First, the input audio signal $x(n)$ of each sung word is sliced into a number of small windows or frames whose size is equal to a power of two. Each signal window is calculated by using the short-time Fourier transform (STFT) defined as follows:

$$X(k)=\sum_{n=0}^{N-1} w(n)x(n)\exp\left(-\frac{2\pi kn}{N}\right)$$

in which $k=0,1,\ldots,\ N-1$, where k corresponds to the frequency $f(k)=\left(kfs/N\right)$. Here fs is the sampling frequency in Hertz and $w(n)$ is Hamming time-window given by

$$w(n)=0.54-0.46\cos\left(\frac{\pi n}{N}\right)$$

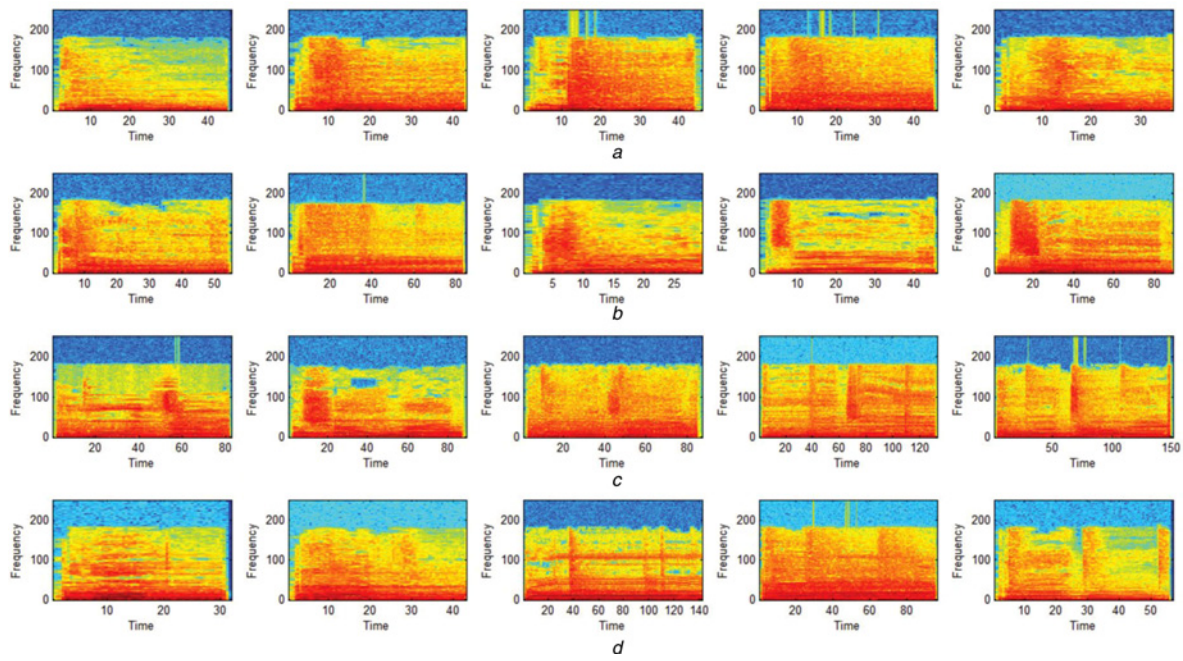

Fig. 1 *Examples of four sung words represented in the form of spectrograms*
A Word 1
B Word 2
C Word 3
D Word 4

The power of each $X(k)$, denoted by $P(k)$, is computed by the following equation:

$$P(k) = 10\log_{10}(X(k))$$

Each $P(k)$ and its time interval are plotted to form a spectrogram feature of each sung word. Fig. 2 shows an example of creating a spectrogram feature. This spectrogram feature is then used as the features of the song and used in the classifying process. In this paper, we used a neural network whose input must be in the form of a vector as a classifier. A power spectrogram feature can be viewed as a collection of columns of power spectrums. Therefore, the spectrogram feature can be transformed into a vector by concatenating the power spectrum columns as shown in Fig. 3.

3.2 *Audio Time Scale Modification (TSM)*

The duration of each sung word in each song is different and depends on both the singer and the tempo. The TSM refers to the process of speeding up or slowing down a sound without changing the pitch of any tonal components. We used three different TSM algorithms to modify the time scales of the audio signals.

Variable Speed Replay or Re-sampling is simplest process to change the duration of a digital audio clip is to re-sample it. Re-sampling is a mathematical operation that effectively rebuilds a continuous waveform from samples of an audio clip and then samples that waveform again at a different rate. When the new samples are played at the original sampling frequency, the audio clip sounds slower or faster similar to changing the speed of an audio tape.

Phase vocoder [57, 58] was developed mainly as a method for compressing speech before transmission. The audio signals are modelled by a set of parameters (e.g., the amplitude and frequencies of the sinusoidal components of the short time segments of the signal) that reproduce the original signal. A Phase Vocoder is a channelised analysis and re-synthesising tool that, through several techniques, measures and stores spectral signal data in different

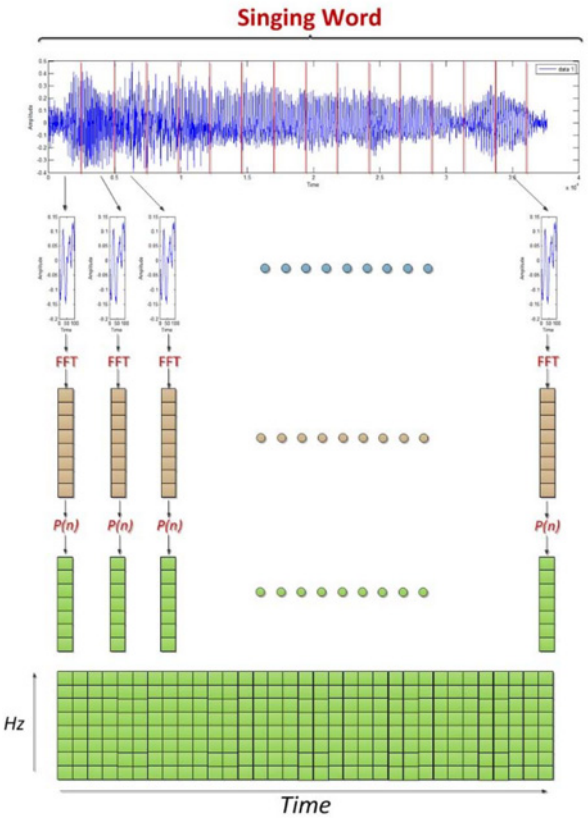

Fig. 2 *Process of computing the power spectrum of an input audio signal and forming the spectrogram feature used in our algorithm*

frequency bands and uses the values to modify and recreate the signal in time domain. Most of the phase vocoder systems are variations on the STFT method for analysis and re-synthesis of a

doi: 10.1049/joe.2017.0210

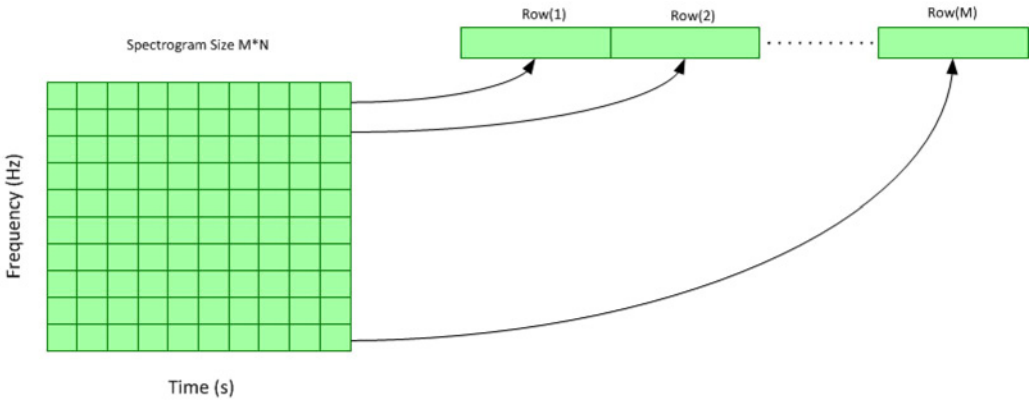

Fig. 3 *Forming the input vector of neural classifier by concatenating columns of power spectrums*

signal. A phase vocoder system based on an STFT technique can be classified into the following main processes: (i) STFT analysis of an input signal, (ii) Modification of parameters, and (iii) STFT synthesis of the output signal. A phase vocoder can shrink or stretch a signal in the time domain without an associated change in pitch. This can be done at the re-synthesis phase by changing the hop size as the new signal is rebuilt.

Waveform Similarity Overlap-and-Add (WSOLA) technique [59, 60] operates in the time domain. The overlap-and-add algorithm is obtained by simply cutting out smoothly windowed chunks of the input audio signal, repositioning them to corresponding time indexes in the output signal, overlapping the windows to obtain continuity, and adding them. WSOLA is a variation of the Fast-Fourier transform similarity in a time-scaled fashion. The excised segments are similar to the adjacent segments. This makes WSOLA a robust time-scaling algorithm that is able to time scale events in the presence of noise or even competing voices in the input audio signal.

4 Data collection

Our system takes a polyphonic music audio signal as its input. The input signals were sampled from CD recordings of music and included musical genres such as pop rock, hard rock, soft rock, dance, hip-hop, Soul, R&B, folk and acoustic. The files were all from different artists. We investigated the performance of a spectrogram feature constructed from audio features to solve the problem of singing voice recognition and provide an empirical evaluation on two datasets.

The dataset was a collection of songs randomly chosen from English and Thai popular music CDs containing over 1500 albums. The details are listed in Table 1. The DB-THS dataset consists of 31 Thai and English one-syllable sung words and greater than or equal to two-syllable sung words and includes 19,200 total sound samples with 600 samples for each word. The 31 considered sung words are shown in Table 2. Each sung word audio sample was selected and manually cut from the songs using the Sony Sound Forge program. All the sample files in Table 1 were coded in stereo at a frequency of 44.2 kHz with a bit rate of 128 kbps.

5 Experimental evaluation

This section discusses the methodology used in our proposed techniques. It includes a description of the experiment setup, the method used for comparisons, and the implementation details. All calculations were done using Matlab 2015a on a desktop computer with an Intel Core i7-4750HQ processor and 16 GB of RAM.

Table 1 The music used in DATASET

Music genres	Male singer	Female singer	Total
pop rock	1768	1545	3313
hard rock	978	667	1645
soft rock	2284	2100	4384
dance	1177	467	1644
hip-pop	304	160	464
soul	250	108	358
R&B	1135	652	1787
folk	297	162	459
acoustic	1288	982	2270
total			16,324

5.1 *Experimental setup*

Our recognition algorithm performed the steps proposed in Section 3. All audio signals were converted to mono and down-sampled at a rate of 22,000 Hz. Each sung word in the dataset was randomly divided into four groups of equal sizes. Then, three randomly selected groups were used for training and the rest were used for testing. Using a cross-validation procedure, the same process was repeated 50 times with the different training and test sets to ensure that all samples were included at least once in the test set. The mean recognition rate was calculated based on the average error for one run on each test set.

6 Results and discussion

In this paper, a three-layer feed-forward network was used for classifying the sounds into the correct sung words as shown in Table 2. A sigmoid transfer function was used in the hidden layer and output layer. The network was trained using a scaled conjugate gradient back-propagation function. The network consists of 31 outputs corresponding to the 31 classes in each dataset. The value of each output is between [0; 1]. The number of hidden neurons was adjusted to achieve the highest accuracy. Although selecting a good learning rule can generate a good result, this paper does not discuss the learning rules because they are not the focus of this study.

6.1 *Selecting the number of neurons in the hidden layer of a neural network*

Determining the number of neurons in the hidden layers is an important aspect in constructing an overall neural network architecture. Though these layers do not directly interact with the external environment, they greatly influence the final output. Both

doi: 10.1049/joe.2017.0210

Table 2 Dataset used in experiments

Class	Singing word	Time duration, s	Pronounce (in Thai)
1	คน	0.65–2.95	'kon'
2	ความ	0.26–0.60	'kwarm'
3	เคย	0.33–0.62	'koey'
4	ใคร	0.33–0.70	'krai'
5	ใจ	0.44–1.38	'jai'
6	ฉัน	0.26–1.23	'chan'
7	ที่	0.26–0.54	'tee'
8	เธอ	0.23–0.78	'ther'
9	มี	0.28–0.86	'mai'
10	รัก	0.18–1.48	'luck'
11	รู้	0.28–0.47s	'roo'
12	เรา	0.26–0.73	'raw'
13	i love you	0.65–2.95	
14	love you	0.57–2.92	
15	together	1.04–2.11	
16	tomorrow	1.07–6.63	
17	yesterday	1.81–5.39	
18	without	0.76–4.90	
19	today	0.81–5.90	
20	the light	0.74–7.91	
21	day go on	1.09–5.54	
22	so far	0.77–6.39	
23	so close	0.65–6.15	
24	be long	0.46–4.21	
25	ความรัก	0.52–3.65	'kwarm-luck'
26	คิดถึง	0.88–1.11	'kit-thun'
27	ใครสักคน	0.99–4.62	'krai-sak-kon'
28	ไม่เคย	0.41–1.99	'mai-koey'
29	ไม่มี	0.57–1.17	'mai-mee'
30	รักเธอ	0.47–1.93	'luck-ther'
31	หัวใจ	0.73–1.46	'hua-jai'

the number of hidden layers and the number of neurons in each of these hidden layers must be carefully considered.

Using too few neurons in the hidden layers will result in inadequately detecting the signals in a complicated dataset, which is called underfitting. Underfitting refers to a model that can neither model the training data nor generalise to new data. An underfit machine learning model is not suitable and will be obvious because it will have low performance on the training data. In contrast, using too many neurons in the hidden layers will result in detecting the signals too strongly in a complicated dataset, which is called overfitting. Overfitting occurs when the neural network has lots of information processing capacity and the limited amount of information contained in the training set is insufficient to train all the neurons in the hidden layers. Therefore, the number of hidden neurons is a relevant factor that affects the accuracy. However, theoretically estimating this number is rather difficult. The criterion to select the parameter value for the number of hidden neurons criterion was selected as follows. Because each sample was captured from a different song with different singers, the duration of each song is different. We applied the Waveform Similarity Based Overlap-Add (WSOLA) to perform TSM of the audio data for each sung word to equalise the lengths of all samples. A duration of 1.0 s was used with a window size of 4096 points with a 25% overlap. We examined the results from a variety of hidden neural unit numbers, used the same setup for each environment type. Three other features, namely, MFCC, LPC and Matching Pursuit (MP) [61], were tested against the spectrogram feature. The parameters of MFCC were the following: the exponent for littering was 0.6, the number of cepstra was 13, the number of warped spectral bands was 40, and the highest band edge of the Mel filters was 4000 Hz.

The frequency warping scale used for filter spacing in MFCC is the Mel scale. For MP, the signal was decomposed using a

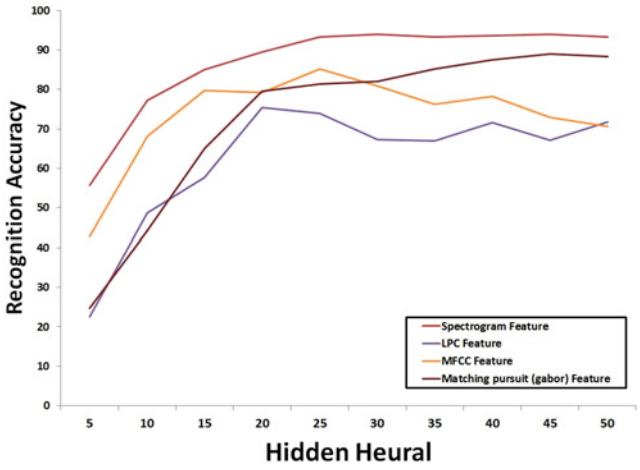

Fig. 4 *Overall recognition accuracy using a feed-forward neural network with varying number of Hidden Neural Unit*

Gabor dictionary of 1200 atoms with dyadic scales ranging from 2 to 256 samples and translations in 0, 64, 128 and 192. For each atom, 35 different exponentially distributed modulation frequencies were considered. From the MP decomposition of the segment, only the first five atoms were used. From these atoms, a four-dimensional feature vector from the mean, standard deviations of the modulation frequencies, and the scales of the five atoms were formed. The classification results for the spectrogram, MP, MFCC and LPC features using a feed-forward neural network with 30 hidden neurons are shown in Fig. 4.

Fig. 4 shows the results from varying the number of hidden neurons and using the same number for each sung word. As the graphs in Fig. 4 show, the spectrogram feature achieved 93.9% accuracy, the highest accuracy among all tested features (i.e., LPC, MFCC, MP and the spectrogram). The resulting recognition rate became constant when using 25 hidden neurons unit over a 25-unit performance at more than 93%. When the number of hidden neurons was increased, the accuracy did not increase much further, which may be due to the overfitting effect during the training process. The highest recognition rate was obtained using 30 hidden neural units, with an accuracy of 93.9%. Thus, we chose to use 25 hidden neural units for the three-layer feed-forward network in our experiments and used that setup to classify all the sounds in all the experiments.

6.2 *Experiments on different TSM algorithms*

The durations of the sung words used in this paper were not equal. Therefore, the size of the feature vectors was not equal either. This study used TSM to solve this problem. TSM equalises the duration of each sound before transforming each sound into a spectrographic image. Many TSM algorithms have been proposed [59, 62, 63]. Here we applied three TSM methods, namely, Variable Speed Replay, Phase Vocoder and WSOLA to equalise the durations of each sound. Figs. 5*A–C* show different spectrographic images of a sung word produced by these different TSM algorithms (WSOLA, Phase Vocoder and Re-sampling, respectively), from different durations. Eleven durations of equal size varying from 0.4 to 1.4 s (e.g., 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3 and 1.4 s) were considered. Notice that the different time scales and intervals yield different informative details. Each input audio signal for each sung word is sliced into several small windows or frames using 4096 points with a 25% overlap. From the discussions in previous sections, the spectrogram feature provided the best performance. Hence, this feature was used along with a 30 hidden-neuron feed-forward network to compare the performance of each TSM algorithm. The results are shown in Fig. 6. From these results, it can

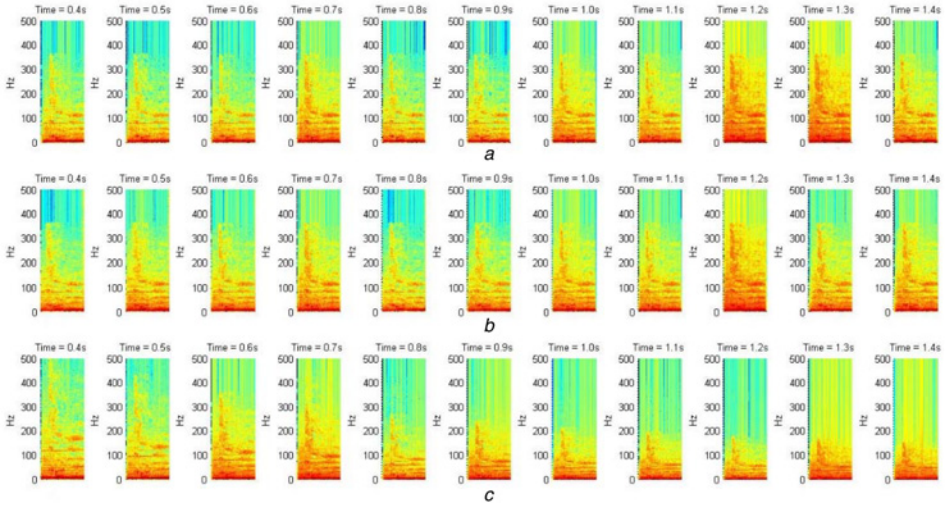

Fig. 5 *Examples of an audio signal spectrogram modified by different time-scale modification algorithms and different time intervals*
A WSOLA
B Phase Vocoder
C Re-sampling

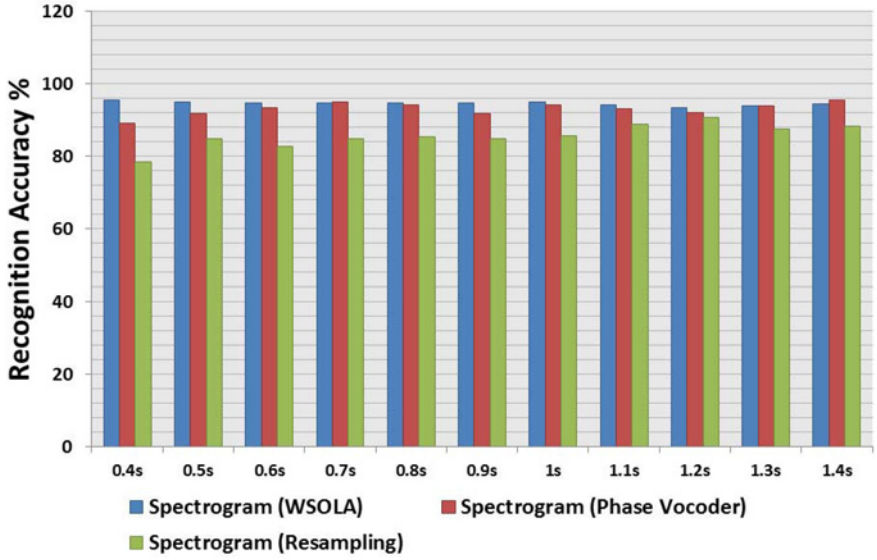

Fig. 6 *Average recognition performance of different time-scale modification algorithms based on a spectrogram with a feed-forward neural network*

be concluded that the WSOLA algorithm when combined with the spectrogram feature provides the best performance among the tested algorithms (Fig. 7).

6.3 *Experiment on different sizes of windowed segment*

A spectrogram can be obtained from different sizes windowed segments. The quantity of information of any sound wave represented in spectrogram form depends on the size of window, as shown in Fig. 8. However, predicting the most suitable window size is not simple. To discover the suitable window size for applying to sung words to achieve the maximum possible classification accuracy, the following set of window sizes (64, 128, 256, 512, 1024, 2048, 4096, 8192) was tested with a neural network. Based on the spectrogram, using MFCC, LPC and MP with different window sizes, Fig. 9 shows the average neural classification accuracy. In these experiments, the neural classifier was configured the same as discussed in the previous section. The following window sizes were tested: 8192, 4096, 2048, 1024, 512, 256 and 128. Two adjacent windows were overlapped by 25% of their width. Fig. 9 show the recognition accuracy of different feature with different window segment sizes (*y*-axis). All the experiments were used 25 hidden neurons for the feed-forward neural network. The spectrogram features provided the highest accuracy compared with MFCC, LPC and MP in most of the experimental cases. For feed-forward networks, a large window size achieves higher accuracy than does a small window size for all features (e.g., spectrogram, MFCC, LPC and MP); however, when the window size is in a range from 512 to 8192, the variations in accuracy are rather narrow. At the maximum window size of 8192, the accuracy of the feed-forward neural network reached 99.97%. This is because a large window contains more information than a small window and the sung word duration of the singing style limited singers and the types of songs.

6.4 *Experiment on different sampling rates*

Recording audio data at a high sampling rate infers that the audio signal will have a higher quality than recording at a lower sampling rate. This is rather obvious because the high sampling rate can capture more details of the signal that may contain the most relevant

doi: 10.1049/joe.2017.0210

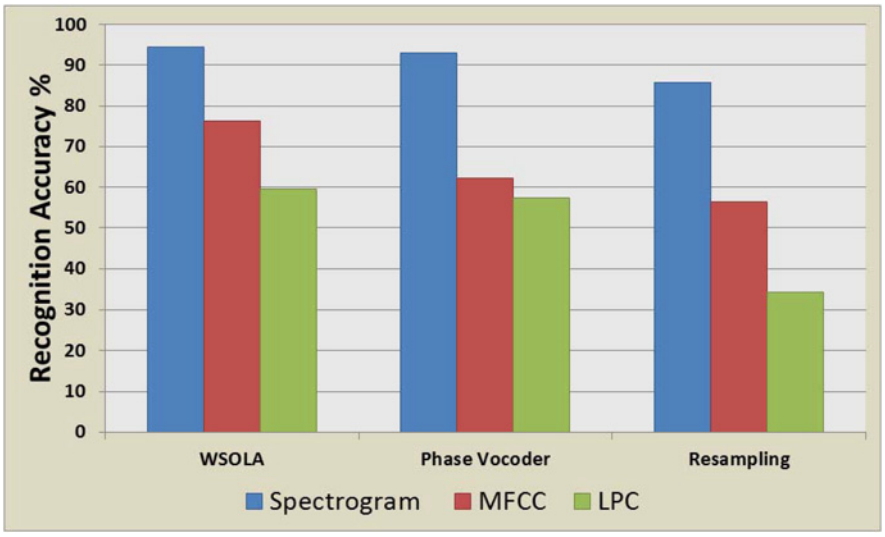

Fig. 7 *Average recognition performance by a feed-forward neural network using the spectrogram, MFCC and LPC features created using the different time-scale modification algorithms for the audio signals shown in Fig. 6*

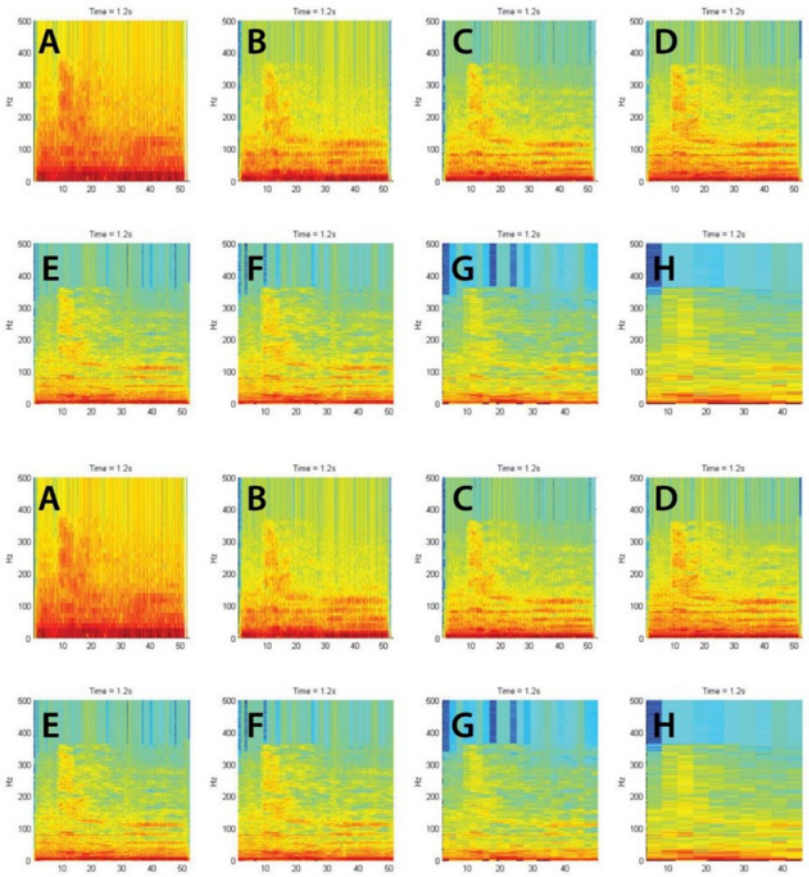

Fig. 8 *Spectrogram examples obtained from different windowed segment sizes*
a 64
b 128
c 256
d 512
e 1024
f 2048
g 4096
h 8192

features for recognition. Fig. 10 shows the spectrogram images with a window size of 512 and with a 25% overlapping segment for a sung word captured at various sampling rates. However, when a higher sampling rate is used, the sound file storage requirements are unnecessarily increased. The problem here is to find the appropriate sampling rate with respect to the specified

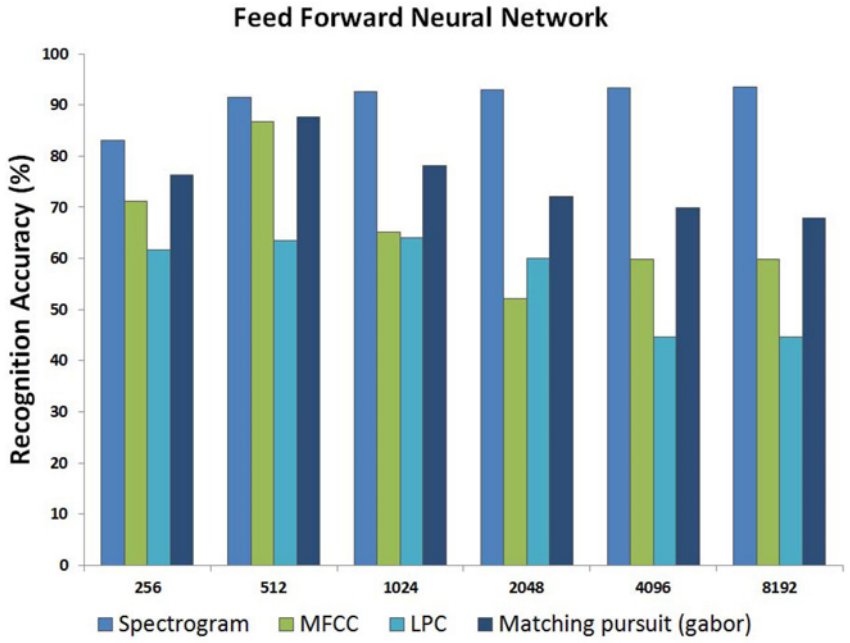

Fig. 9 *A comparison of recognition accuracy for different window sizes based on a 35-hidden-neuron feed-forward neural network using different features*

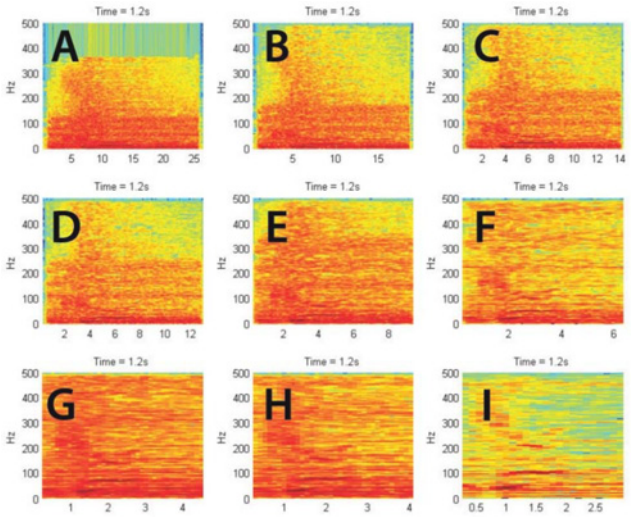

Fig. 10 *Examples of spectrogram images created using different sampling rates*
A 44,100 Hz
B 32,000 Hz
C 24,000 Hz
D 22,050 Hz
E 16,000 Hz
F 11,025 Hz
G 8000 Hz
H 7333 Hz
I 5500 Hz

accuracy. In this experiment, all audio files were coded in stereo at a frequency of 44.2 kHz with a 128 kbps bit rate; then, they were converted to mono and down-sampled to frequencies of 5500, 6000, 7333, 8000, 11,025, 16,000, 22,050, 32,000 and 44,100 Hz. The same experimental settings used in the previous section were deployed here.

Fig. 11 shows the accuracy resulting from different sampling rates using the feed-forward neural network. The spectrogram feature yields the highest accuracy except at frequencies between 22,050 and 44,100 Hz.

6.5 *Comparison with other classification techniques*

The following classification techniques are used for speech/voice recognition or have, in the past, been used for this purpose:

- KNN
- Fishers Linear Classifier
- Linear Bayes Normal Classifier
- Naive Bayes Classifier
- Parzen Classifier
- Decision Tree

In this experiment, we used the data from Table 2. Based on our-experimental setup, we used a window of 4096 pixels with a 25% overlap for all feature extractions. Then, we applied WSOLA algorithms to resize the length of the spectrogram feature to 1.0 s. We compared the overall recognition accuracy using the spectrogram features and their combinations for the 32 classes of sung words in Table 2 with the seven classification technique shown in Fig. 12. The results of varying the classification techniques in the graphs in Fig. 12, show that the spectrogram feature achieved 93.9% accuracy, the highest among all the tested classification techniques.

6.6 *Comparison with the ASR algorithm*

An interesting benchmark is shown in Fig. 13, we ran the same experiments using the spectrogram features and compared our approach with the ASR algorithm. With the ASR algorithm, we used the HMM with the same data but with the LPC and MFCC 13 coefficients. To compare our algorithm with the ASR algorithm, each sung word in Table 2 was randomly divided into four groups of equal sizes. Then, we arbitrarily selected three groups for training and used the rest for testing. For the cross-validation procedure, the same process was repeated 50 times with different training and test sets, ensuring that all the samples were included at least once in the test set. The mean recognition rates were calculated based on the average error for one run on each test set. For our algorithm, we used windows of 4096 to create a spectrogram feature. A spectrogram feature was provided to the feed-forward neural network. We used 25 hidden neurons

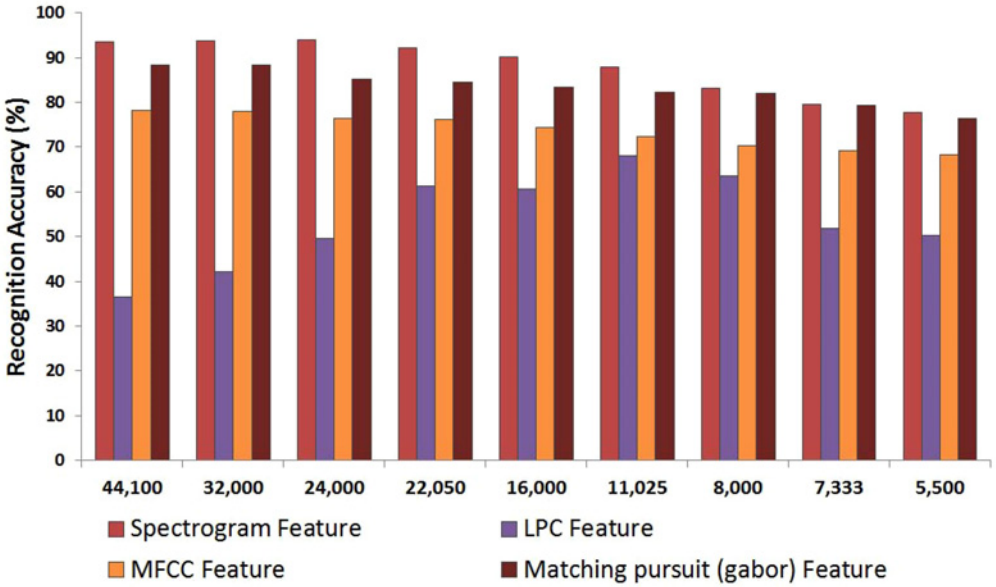

Fig. 11 *Average recognition accuracy from different sampling rates based on the spectrogram, MFCC and LPCC features with a feed-forward neural network*

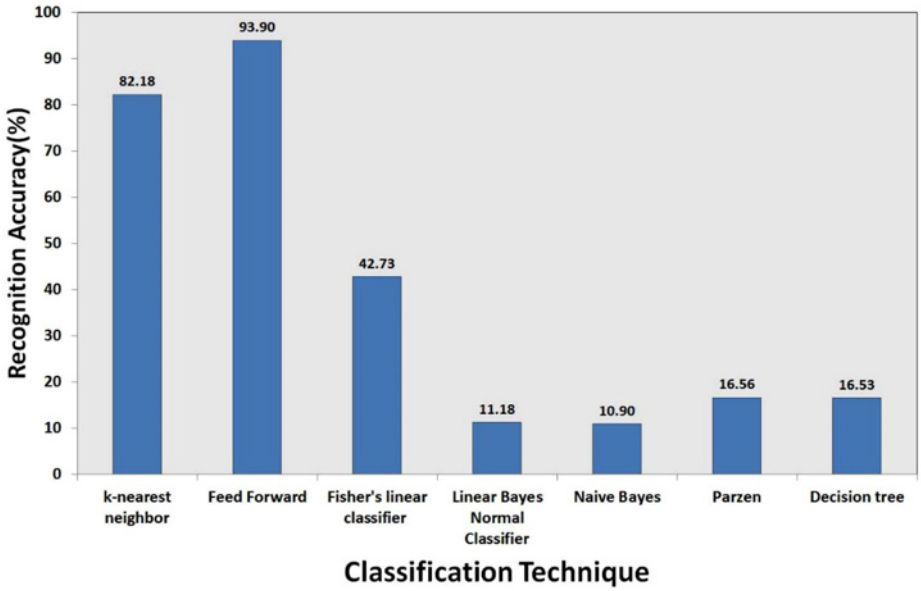

Fig. 12 *Accuracy results from different classification techniques using the spectrogram feature on the DB-TH-ENG Dataset*

for all the tests. To compare the experimental results with the ASR algorithm, we used an HMM and the same data but with the LPC and MFCC 13 coefficients.

The results presented Fig. 13 show the overall recognition accuracy comparing the spectrogram features with the feed-forward neural network to the ASR algorithm for the full sound dataset. As shown, the spectrogram features achieved the highest recognition rate 94.9%. This combination performs better than the ASR algorithm for the full dataset. Therefore, it seems likely that recognising sung words is quite different from recognising spoken text and that the ASR regards the background instrumental accompaniment as noise, which degrades its performance. From this section, it is clear that the spectrogram feature in combination with a feed-forward neural network can solve the singing voice recognition problem. Notably, the spectrogram feature recognises the cross-language music data listed in Table 2 without using any method to separate the music from the background.

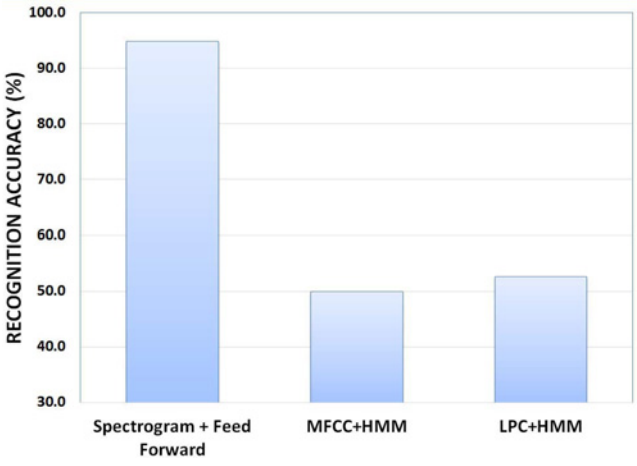

Fig. 13 *Overall recognition rate*

7 Conclusion

In this paper, we proposed an algorithm for singing voice recognition from monaural polyphonic music based on spectrogram images and a neural network classifier, an image resizing algorithm and classification algorithms. However, a spectrogram is also limited. The dimensions of spectrogram features are very high and the time interval of each sung word is not equal. Consequently, we applied image-resizing algorithms to solve both problems. The results show that all the tested classifiers can recognise a sung word even when it is superimposed over background music. The experiment showed that the feed-forward neural network performed better than the ASR, achieving an accuracy rate of 93.90%. Notably, the algorithm can recognise cross-language music data.

8 Acknowledgements

This work was supported by the Thailand Research Fund (TRF) under Grant Number TRG5780202. The support is gratefully acknowledged.

9 References

[1] Cullity B.D.: 'Music information retrieval', vol. 35 (Information Today Books, 2003)
[2] Hayashi T., Ishii N., Yamaguchi M.: 'Fast music information retrieval with indirect matching'. 2014 Proc. 22nd European Signal Processing Conf. (EUSIPCO), September 2014, pp. 1567–1571
[3] Vaizman Y., McFee B., Lanckriet G.: 'Codebook-based audio feature representation for music information retrieval', *IEEE/ACM Trans. Audio Speech Lang. Process.*, 2014, **22**, (10), pp. 1483–1493
[4] McVicar M., Santos-Rodriguez R., Ni Y., *et al.*: 'Automatic chord estimation from audio: a review of the state of the art', *IEEE/ACM Trans. Audio Speech Lang. Process.*, 2014, **22**, (2), pp. 556–575
[5] Su L., Yeh C.-C.M., Liu J.-Y., *et al.*: 'A systematic evaluation of the bag-of-frames representation for music information retrieval', *IEEE Trans. Multimed.*, 2014, **16**, (5), pp. 1188–1200
[6] Raposo F., Ribeiro R., Martins de Matos D.: 'Using generic summarization to improve music information retrieval tasks', *IEEE/ACM Trans. Audio Speech Lang. Process.*, 2016, **24**, (6), pp. 1119–1128
[7] Shetty S., Hegde S.: 'Clustering of instruments in Carnatic music for content based information retrieval'. 2016 IEEE 6th Int. Conf. on Advanced Computing (IACC), February 2016, pp. 127–132
[8] Dridi A., Kacimi M.: 'Kiss mir: Keep it semantic and social music information retrieval'. 2015 Seventh Int. Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), vol. 01, November 2015, pp. 433–439
[9] Gerhard D.B.: 'Computationally measurable differences between speech and song'. PhD thesis, Burnaby, BC, Canada, Canada, 2003, AAINQ81587
[10] Sasou A., Goto M., Hayamizu S., *et al.*: 'An auto-regressive, non-stationary excited signal parameter estimation method and an evaluation of a singing-voice recognition'. Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing, 2005 (ICASSP '05), vol. 1, 18–23 2005, pp. 237–240
[11] Yaguchi Y., Oka R.: 'Song wave retrieval based on frame-wise phoneme recognition', in Lee G., Yamada A., Meng H., Myaeng S. (eds.): 'Information retrieval technology' (Springer, Berlin/Heidelberg, 2005) (*LNCS*, **3689**), pp. 503–509
[12] Tsai W.H., Lee H.C.: 'Singer identification based on spoken data in voice characterization', *IEEE Trans. Audio Speech Lang. Process.*, 2012, **20**, (8), pp. 2291–2300
[13] Duan Z., Fang H., Li B., *et al.*: 'The nus sung and spoken lyrics corpus: a quantitative comparison of singing and speech'. 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. (APSIPA), October 2013, pp. 1–9
[14] Loni D.Y., Subbaraman S.: 'Formant estimation of speech and singing voice by combining wavelet with lpc and cepstrum techniques'. 2014 Ninth Int. Conf. on Industrial and Information Systems (ICIIS), December 2014, pp. 1–7
[15] Chan P.Y., Dong M., Lim Y.Q., *et al.*: 'Formant excursion in singing synthesis'. 2015 IEEE Int. Conf. on Digital Signal Processing (DSP), July 2015, pp. 168–172
[16] Li X., Wang Z.: 'A hmm-based mandarin Chinese singing voice synthesis system', *IEEE/CAA J. Autom. Sin.*, 2016, **3**, (2), pp. 192–202
[17] Makeyev O., Sazonov E., Schuckers S., *et al.*: 'Limited receptive area neural classifier for recognition of swallowing sounds using short-time Fourier transform'. Int. Joint Conf. on Neural Networks, 2007 (IJCNN 2007), August 2007, pp. 1601–1606
[18] Lin C.-C., Chen S.-H., Truong T.-K., *et al.*: 'Audio classification and categorization based on wavelets and support vector machine', *Speech and Audio Processing, IEEE Transactions on*, 2005, **13**, (5), pp. 644–651
[19] Esmaili S., Krishnan S., Raahemifar K.: 'Content based audio classification and retrieval using joint time-frequency analysis'. Proc. Int. Conf. on Acoustics, Speech, and Signal Processing, 2004 (ICASSP'04)., vol. 5, May 2004, pp. 665–668
[20] Wang J.-C., Lee H.-P., Wang J.-F., *et al.*: 'Robust environmental sound recognition for home automation', *IEEE Trans. Autom. Sci. Eng.*, 2008, **5**, (1), pp. 25–31
[21] Yoshii K., Goto M., Okuno H.G.: 'Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression', *IEEE Trans. Audio Speech Lang. Process.*, 2007, **15**, (1), pp. 333–345
[22] Toyoda Y., Huang J., Ding S., *et al.*: 'Environmental sound recognition by the instantaneous spectrum combined with the time pattern of power'. Neural Networks and Computational Intelligence, 2004, pp. 169–172
[23] Makeyev O., Sazonov E., Schuckers S., *et al.*: 'Limited receptive area neural classifier for recognition of swallowing sounds using continuous wavelet transform'. 29th Annual Int. Conf. of Engineering in Medicine and Biology Society, 2007 (EMBS 2007), August 2007, pp. 3128–3131
[24] Ajmera J., McCowan I., Bourlard H.: 'Speech/music segmentation using entropy and dynamism features in a hmm classification framework', *Speech Commun.*, 2003, **40**, pp. 351–363
[25] Toyoda Y., Huang J., Ding S., *et al.*: 'Environmental sound recognition by multilayered neural networks'. Fourth Int. Conf. on Computer and Information Technology, 2004 (CIT '04), September 2004, pp. 123–127
[26] Shenoy A.: 'Singing voice detection for karaoke application', *Proc. SPIE*, 2005, **5960**, pp. 752–762
[27] Nwe T.L., Shenoy A., Wang Y.: 'Singing voice detection in popular music'. Proc. 12th Annual ACM Int. Conf. on Multimedia (MULTIMEDIA '04), New York, NY, USA, 2004, pp. 324–327
[28] Tsai W.-H., Wang H.-M., Rodgers D., *et al.* 'Blind clustering of popular music recordings based on singer voice characteristics'. ISMIR, 2003
[29] Berenzweig A.L., Ellis D.P.W.: 'Locating singing voice segments within music signals'. 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2001, pp. 119–122
[30] Chou W., Gu L.: 'Robust singing detection in speech/music discriminator design'. Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 2000., Washington, DC, USA, 2001, vol. 02, pp. 865–868
[31] Berenzweig A.L., Ellis D.P.W., Lawrence S.: 'Using voice segments to improve artist classification of music'. Audio Engineering Society Conf.: 22nd Int. Conf. on Virtual, Synthetic, and Entertainment Audio, vol. 6, 2002
[32] Maddage N.C., Xu C., Wang Y.: 'An SVM-based classification approach to musical audio'. ISMIR, 2003
[33] Maddage N.C., Wan K., Xu C., *et al.*: Singing voice detection using twice-iterated composite Fourier transform'. 2004 IEEE Int. Conf. on Multimedia and Expo, 2004 (ICME '04), vol. 2, June 2004, pp. 1347–1350
[34] Rocamora M., Herrera P.: 'Comparing audio descriptors for singing voice detection in music audio files'. Eleventh Brazilian Symp. on Computer Music, San Pablo, Brazil, September 2007
[35] Tzanetakis G.: 'Song-specific bootstrapping of singing voice structure'. 2004 IEEE Int. Conf. on Multimedia and Expo, 2004 (ICME '04), vol. 3, June 2004, pp. 2027–2030
[36] Kim Y.E.: 'Singer identification in popular music recordings using voice coding features'. Proc. of the Third Int. Conf. on Music Information Retrieval, 2002, pp. 164–169
[37] Suzuki M., Hosoya T., Ito A., *et al.*: 'Music information retrieval from a singing voice using lyrics and melody information', *EURASIP Appl. J. Signal Process.*, 2007, **2007**, pp. 151–151
[38] Wong C., Szeto W., Wong K.: 'Automatic lyrics alignment for Cantonese popular music', *Multimed. Syst.*, 2007, **12**, (4/5), pp. 307–323
[39] Kan M.-Y., Wang Y., Iskandar D., *et al.*: 'Lyrically: Automatic synchronization of textual lyrics to acoustic music signals', *IEEE Trans. Audio Speech Lang. Process.*, 2008, **16**, (2), pp. 338–349

[40] Gruhne M., Schmidt K., Dittmar C.: 'Phoneme recognition in pop-pular music'. Eighth Int. Conf. on Music Information Retrieval, Vienna, Austria, 23–27 September 2007, pp. 2027–2030
[41] Fujihara H., Goto M., Ogata J., *ET AL*. 'Automatic synchronization between lyrics and music cd recordings based on viterbi alignment of segregated vocal signals'. Proc. of the Eighth IEEE Int. Symp. on Multimedia (ISM '06), Washington, DC, USA, 2006, pp. 257–264
[42] Zwan P., Szczuko P., Kostek B., *ET AL*.: 'Transactions on rough sets ix Chapter'. Automatic Singing Voice Recognition Employing Neural Networks and Rough Sets, 2008, pp. 455–473
[43] Mesaros A., Virtanen T.: 'Recognition of phonemes and words in singing'. 2010 IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP), March 2010, pp. 2146–2149
[44] Hu Y., Liu G.: 'Singer identification based on computational auditory scene analysis and missing feature methods', *Intell. J.: Inf. Syst.*, 2014, **42**, (3), pp. 333–352
[45] Raj B.: 'Separating a foreground singer from background music'. Int. Symp. on Frontiers of Research on Speech and Music, 2007
[46] Huang P.-S., Chen S.D., Smaragdis P., *ET AL*.: 'Singing-voice separation from monaural recordings using robust principal component analysis'. 2012 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), March 2012, pp. 57–60
[47] Chanrungutai A., Ratanamahatana C.A.: 'Singing voice separation for mono-channel music using non-negative matrix factorization'. 2008 Int. Conf. on Advanced Technologies for Communications, October 2008, pp. 243–246
[48] Zhu B., Li W., Li R., *ET AL*.: 'Multi-stage non-negative matrix factorization for monaural singing voice separation', *IEEE Trans. Audio Speech and Lang. Process.*, 2013, **21**, (10), pp. 2096–2107
[49] Durrieu J.L., David B., Richard G.: 'A musically motivated mid-level representation for pitch estimation and musical audio source separation', *IEEE J. Sel. Top. Signal Process.*, 2011, **5**, (6), pp. 1180–1191
[50] imekli U., Cemgil A.T.: 'Score guided musical source separation using generalized coupled tensor factorization'. 2012 Proc. 20th European Signal Processing Conf. (EUSIPCO), August 2012, pp. 2639–2643
[51] Mohammadiha N., Smaragdis P., Leijon A.: 'Supervised and unsupervised speech enhancement using nonnegative matrix factorization', *IEEE Trans. Audio Speech Lang. Process.*, 2013, **21**, (10), pp. 2140–2151
[52] Yoo J., Kim M., Kang K., *ET AL*.: 'Nonnegative matrix partial co-factorization for drum source separation'. 2010 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, March 2010, pp. 1942–1945
[53] Kim M., Yoo J., Kang K., *ET AL*.: 'Nonnegative matrix partial co-factorization for spectral and temporal drum source separation', *IEEE J. Sel. Top. Signal Process.*, 2011, **5**, (6), pp. 1192–1204
[54] Ono N., Miyamoto K., Le Roux J., *ET AL*.: 'Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram'. 2008 16th European Signal Processing Conf., August 2008, pp. 1–4
[55] Goto M.: 'A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals', *Speech Commun.*, 2004, **43**, (4), pp. 311–329, Special Issue on the Recognition and Organization of Real-World Sound
[56] Ikemiya Y., Itoyama K., Yoshii K.: 'Singing voice separation and vocal f_0 estimation based on mutual combination of robust principal component analysis and subharmonic summation', *IEEE/ACM Trans. Audio Speech Lang. Process.*, 2016, **24**, (11), pp. 2084–2095
[57] Gournay P., Lefebvre R., Savard P.-A.: 'Hybrid time-scale modification of audio'. Audio Engineering Society Convention, 2007, vol. 122, p. 5
[58] Laroche J., Dolson M.: 'Improved phase vocoder time-scale modification of audio', *IEEE Trans. Speech Audio Process.*, 1999, **7**, (3), pp. 323–332
[59] Verhelst W., Roelands M.: 'An overlap-add technique based on waveform similarity (Wsola) for high quality time-scale modification of speech'. Proc. 1993 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing: Speech Processing – Volume II (ICASSP'93), Washington, DC, USA, 1993, pp. 554–557
[60] Grofit S., Lavner Y.: 'Time-scale modification of audio signals using enhanced Wsola with management of transients', *IEEE Trans. Audio Speech Lang. Process.*, 2008, **16**, (1), pp. 106–115
[61] Chachada S., Jay Kuo C.-C.: 'Environmental sound recognition: a survey', *APSIPA Trans. Signal Inf. Process.*, 2014, **3**, p. 001
[62] Grofit S., Lavner Y.: Time-scale modification of audio signals using enhanced wsola with management of transients', *IEEE Trans. Audio Speech Lang. Process.*, 2008, **16**, (1), pp. 106–115
[63] Ninness B., Henriksen S.J.: 'Time-scale modification of speech signals', IEEE Trans. Signal Process., 2008, **56**, (4), pp. 1479–1488