Bio-acoustic signal identification based on sparse representation classifier

D A Ramli, Z X Wan and H Ibrahim
School of Electrical & Electronic Engineering, USM Engineering Campus, Universiti Sains Malaysia, 14300, Nibong Tebal, Penang, Malaysia
dzati@usm.my

Abstract. Most animals produce sounds as a way of communication within their species or as noises resulting from feeding or travelling. An automated recognition of bio-acoustic signals becomes vital in the aspect of ecological research and environmental monitoring. With the improvement of technology, scientists today are interested in classifying types and species of animals by their vocalizations without visualizing them with the naked eye. Hence, species identification system based on animal vocalizations becomes an important topic to be researched nowadays. This project aims to develop a frog species voice identification system, recognizing different frog species through analyzing their calls. In this project, Sparse Representation Classifier (SRC) and Kernel Sparse Representation Classifier (KSRC) are employed for the identification task. Performances between SRC and KSRC are compared and discussed in this project. Besides, a graphical user interface (GUI) is also developed to facilitate the user while interacting with the system. Two experiments were done in this project so as to evaluate the effect of numbers of training data and feature dimensions to the classifiers. In short, KSRC (96.6667%) has a higher performance in accuracy compared to SRC (95.6667%). However, KSRC takes a longer computation time compared to SRC. A GUI namely Frog Species Identifier that has been developed by implementing KSRC with 20 training data and 4096 feature dimensions is the final outcome of this project.

1. Introduction
Recognition of animal species is common and important nowadays. Modern technology allows the recognition process becomes feasible and efficient. The most typical animal recognition technique is normally based on the appearance of the animal itself. Recent developments in wildlife monitoring tools by radio tracking, wireless sensor network tracking, global positioning system (GPS) tracking and motion-sensitive camera traps are some example where the animal appearance is used as the recognition modality [1]. Apart from that, animal species can also be recognized through DNA sequence variation [2].

Animal species recognition based on animal’s calls or sounds is another progressive technique to identify animal species [3]. For animals, initiation of sound can be meant of information transfer or can be the noise they made while travelling or feeding. Most animals produce sound to communicate with their own species. However, some sound produced may not fall in the hearing frequency of human being [3]. The advantage of using sound approach for animal species recognition is it can be executed from distance as the animals may live in secluded area.
At the moment, animal sound recognition for analysing the habits and distribution of animals at certain environment has become an active study. The purpose of this study is to monitor and to improve the survivability of the animals [3]. In recent years, numbers of published researches on the development of machine learning algorithm for automatic animal call recognition system have increased. Not only land animals are in the researchers’ interest, marine mammals’ sound classification has also been reported as well [4]. For example in this study, a feature extraction based on octave analysis was performed to capture various sounds in the ocean that can allow marine scientist to detect, identify and locate endangered marine species [4]. In another field, a research on bird species identification via transfer learning from music genres was reported [5]. In this research, transfer learning is proposed to identify bird species based on the existing acoustic similarities.

Apart from the marine mammal and bird species sound recognition, frog species recognition by frog call can also be explored. The motivation of developing a frog species identifier is because amphibians play a vital role in our environment and ecosystem. Certain species of amphibians are useful as indicators of ecosystem stress. Normally, environmental stress is defined as the biological, chemical and physical constraints on the productivity of species and development of ecosystems. When the exposure to environmental stressors increases or decreases in intensity, it infers the ecological responses [6]. Therefore, an intelligent system that can facilitate the effort to estimate frog community calling activity and species richness has been developed by [7]. A frog call biometric identification system for recognizing frog species has also been developed by [8]. In this study, frog calls were first processed into salient features and classified using support vector machine (SVM) technique [8].

This paper focuses on development of frog calls biometric identification system using automated Sparse Representation Classifier (SRC). For the past decades, many classification approaches such as Gaussian mixture model (GMM), Hidden Markov Model (HMM), Support Vector Machine (SVM) have been successfully applied [2],[3]. However, study and development of SRC have now become prevailing in recent years. Sparse representation of signal can be expressed by a linear combination of atoms in an over-complete dictionary in which some of the entries are non-zero [9],[10],[11],[12]. In this paper, the accuracy of SRC and the implementation of kernel to SRC algorithm will be observed and verified in the perspective of frog sound species identification.

The rest of this paper is organized as follows. Section 2 summarizes the related theory on SRC. Section 3 describes the methodology. Finally, Section 4 presents the experimental results and discussions.

2. Related theory on Sparse Representation Classifier (SRC)

Sparse Representation is commonly used for classification [9],[10],[11],[12]. In mathematical terms, a linear combination of SRC can be written as Equation 1 below:

\[ y_{m \times 1} = A_{m \times n} x_{n \times 1} \]  \hspace{1cm} (1)

where \( y \) is the input signal in \( R^{m \times 1} \) space, \( A \) is the dictionary in \( R^{m \times n} \) space and \( x \) is the sparse solution in \( R^{n \times 1} \) space. In order to ensure that \( x \) is sparse, i.e. many of its entries are zero, optimization problem involving Basis Pursuit (BP) and Matching Pursuit (MP) to replace the \( l_0 \) norm with \( l_1 \) norm is used in solving the minimization. In this project, we will be going to train and develop the dictionary \( A \) to identify the input data that provides an accurate sparse solution. Frog calls which are in signal wave form will be taken as data to undergo feature extraction module before further classified by SRC for identification.

However, if the sparse representation classifier is to be implemented in real time application, the elapsed time for solving classification is one of the disadvantages. This problem is mainly caused by the sparse signal solver, which is \( l_1 \) minimization or Basis Pursuit [13]. Hence, this research is done to improve the efficiency of the sparse signal recovery solver. In this work, a smooth \( l_0 \) norm solver is
modified and implemented to improved accuracy of classification apart from reducing computation time. Kernel sparse representation as a modified version to this solver is further described in this paper.

3. Methodology

Entire process involved for the identification system development is described in this section. The frog sound identification system module consists of 3 sub-modules, which are data acquisition module, feature extraction module and lastly the classification module. Figure 1 below shows the overview of the methodology.

![Figure 1](image)

**Figure 1.** Overview of identification system development.

3.1 Data Acquisition and pre-processing

In this project, the digital frog call samples are obtained from the database collected by Intelligent Biometric Research Group (IBG), School of Electrical and Electronic Engineering, Universiti Sains Malaysia. The frog calls were collected from two different locations of forest in Kedah, Malaysia. The time frame to obtain the raw data of the frog calls is between February 2012 and July 2013. The first location of data collection was done near Sungai Sedim, Kulim. The sounds were recorded beside a river at 8.00pm to 12.00am. Another location is Baling, Kedah in which the frog calls were collected in a swamp area between 6.00pm to 10.00pm. The frog calls were recorded by using a Sony Stereo IC
Recorder ICD-AX412F together with an electric condenser microphone of 32 kHz sampling frequency with WAV format. The sound samples were then converted to 16-bit mono. Finally, each frog sample is identified and labelled by scientist, comprising of 15 labelled frog species.

A Band-pass filter is used to minimise noise or unwanted sounds in the recorded audio. By using the band-pass filter, the region of interest (bandwidth) can be preserved while the rest will be filtered. A call that a frog produces with a single blow of air from its lungs is called as syllable. A set of features can be calculated to represent each syllable once the syllables have been property segmented. Then, the pre-processing step which involved pre-emphasis, framing and windowing (Hamming) will take place before the feature extraction process.

3.2 Feature Extraction by MFCC

MFCC is commonly used as a function for feature extraction, especially in automated speech recognition and speaker recognition. The computation of MFCC is based on short-time Fourier Transform (STFT) analysis. The 4-step pseudo code to implement MFCC is as below:

i. Discrete Fourier Transform (DFT) of all pre-processed signals are computed from the output of windowing, $x_t(n)$. The DFT of all frames of the signals is

$$x_t(n) = X_t(e^{\frac{2\pi k}{N}}), k = 0,1,2,..., N - 1$$

where $\frac{2\pi k}{N} = \omega$.

ii. Next, the signal spectrum is processed by filter bank processing. Filter bank is a set of 24 band-pass filter which emphasizes on the processing spectrum which is below 1kHz. The $m$-th filter bank output is:

$$Y_t(m), 1 \leq m \leq M$$

where $M$ is the number of band pass filter.

iii. Lastly, the inverse DFT is performed on the logarithm of the magnitude of the filter bank output. The output equation is as below.

$$y_t^{(m)}(k) = \sum_{m=1}^{m} \log_{10}(|Y_t(m)|) \cdot \cos \left( k \left( m - \frac{1}{2} \right) \frac{\pi}{M} \right), k = 0, ..., L$$

$k$ is the number of cepstral coefficients excluding the 0-th coefficient and $k = 12$.

3.3 Classification by SRC and KSRC

By referring equation (1), SRC implementation is summarized as below:

i. The input for SRC is a matrix of dictionary constructed of training samples.

$$A = [A_1, A_2, ..., A_c] \in R^{m \times n}$$

For $c$ classes, a test sample $y \in R^m$ and an optional error tolerance $\epsilon > 0$.

ii. The atoms of $A$ are normalized to have unit $l_2$ norm to make the model more sensitive to errors.

iii. Next, $l_1$ minimization problem is solved using SPGL1 Toolbox

iv. Compute the residuals, $r_i(y) = ||y - A \xi x||_2$ for $i = 1, ..., c$

v. The class, $d$ of the given test sample, $y$ is determined by

$$d = \text{identity } (y) = \min_i r_i(y)$$

Subsequently, KSRC implementation is summarized as below:

i. The input for KSRC is a matrix of dictionary constructed of the training samples.

$$A = [A_1, A_2, ..., A_c] \in R^{m \times n}$$

For $c$ classes, a test sample $y \in R^m$ and an optional error tolerance $\epsilon > 0$.

ii. Kernelizing $A$ and $y$ to yield $A_{\text{kernel}}$ and $y_{\text{kernel}}$.

iii. The atoms of $A_{\text{kernel}}$ and $y_{\text{kernel}}$ are normalized to have unit $l_2$-norm

iv. Next, $l_1$ minimization is solved using SPGL1 Toolbox

v. Compute the residuals $r_i(y_{\text{kernel}}) = ||y_{\text{kernel}} - A_{\text{kernel}} \xi x||_2$ for $i = 1, ..., c$. 
vi. The class, \( d \) of the given test sample, \( y \) is determined by

\[
d = \text{identity}(y_{\text{kernel}}) = \min_i r_i(y_{\text{kernel}})
\]

### 3.4 Performance Evaluation

To evaluate the performance of SRC and KSRC, 15 species of frogs with 30 frog calls for each species were divided into two sets of data, the training samples and testing samples. Two experiments have been conducted to evaluate the performance of each classifier as below:

i. Performance of SRC and KSRC based on different numbers of training samples.

ii. Performance of SRC and KSRC based on different feature dimension sizes.

After the classification process of SRC and KSRC is done, the accuracy of the classifiers is computed by using the accuracy formula defined below:

\[
\text{Accuracy} \, (\%) = \frac{N_c}{N_s} \times 100\%
\]

where \( N_c \) is the number of correctly recognized class (species), and \( N_s \) is the total number of test samples. In this study, the computation time (elapsed time) of both classifiers in both experiments is recorded and compared as well.

From the performance evaluation procedure, the number of training samples (Experiment 1) and dimension of features (Experiment 2) that achieve the highest accuracy is recorded and will be used for the implementation of the graphic user interface (GUI) of the frog species identifier.

### 3.5 Graphic User Interface (GUI)

A graphic user interface (GUI) is programmed using MATLAB software and serves as an end-product for users. In general, the programmed GUI is used to identify the species of frog sound in real time. The user interface is created by showing the frog profile, i.e. frog image, frog call audio, general name, scientific name and also a short description regarding the identified species.

### 4. Results and discussions

In this section, performances of the frog call identification system using SRC and KSRC are discussed.

#### 4.1 Performance of SRC and KSRC based on different number of training samples

Table 1 shows the accuracy and elapsed time of SRC and KSRC based on different numbers of training samples. In the experiment, 10, 15 and 20 training samples were tested for both SRC and KSRC. The numbers of test samples were fixed at 10 samples and the numbers of frog species (classes) were also fixed for 15 classes.

| Number of Training Samples | SRC          | KSRC         |
|----------------------------|--------------|--------------|
|                            | Accuracy (%) | Accuracy (%) |
|                            | Elapsed Time (s) | Elapsed Time (s) |
| 10                         | 81.330       | 85.330       |
|                            | 37.956       | 39.090       |
| 15                         | 82.000       | 88.000       |
|                            | 38.597       | 40.188       |
| 20                         | 87.3333      | 94.6667      |
|                            | 43.865       | 45.144       |

From Table 1, both accuracy of SRC and KSRC improves with the increase of the number of training samples. As the numbers of training samples increases, hence more salient information is used by the algorithm to develop the dictionary. The results also revealed that KSRC always achieves higher accuracy compared to SRC. KSRC achieves the highest accuracy of 94.6667% with 20 training samples. This observation can be explained as the nonzero entries of sparse representation, \( x \) of the
test sample is more associated with training samples from same class itself. Apart of that, another factor that improves the classification accuracy of KSRC is that the non-linear feature space input is kernelized into a linear higher-dimensional kernel feature space. KSRC showed very convincing results which have proved its ability in classification application not only on image data but also in sound based classification as well.

Based on the results in Table 1, the computation time of the classifier takes longer time as the number of training sample increases. The increase of the number of atoms in the dictionary thus causing the sparse code dimension increases. Hence, longer computation time is needed for both classifiers to compute the sparse solution. Therefore, the elapsed time of SRC increases from 37.956s to 38.597s and subsequently to 43.865s when the training samples increase from 10 to 15 and 15 to 20 of training samples, respectively. The same trend is also observed for KSRC. The computation time increases from 39.090s to 40.188s and subsequently to 45.144s when the training samples increase from 10 to 15 and 15 to 20 of training samples, respectively.

Even though KSRC has a better performance than SRC in term of accuracy, the computation time of KSRC is the main trade-off of this algorithm for being implemented in real time application. From Table 1, the time taken to complete the classification process of KSRC is longer compared to SRC. This is because in KSRC, when kernel trick is applied into the classifier algorithm, the distribution of samples needs to be transformed by mapping the classifier into a high dimensional kernel feature space. This procedure is done by changing the linear inseparable samples in the original feature space into linear separable in the high dimensional feature space. By that, a test sample can be represented as the linear combination of training samples from the same class more accurately by applying kernel trick into the SRC algorithm.

4.2 Performance of SRC and KSRC based on different feature dimension sizes

Table 2 shows the accuracy and elapsed time of classifiers based on different feature dimension of frog call samples.

| Feature Dimensions | SRC   | KSRC  |
|--------------------|-------|-------|
| 2401 (49*49)       | 91.333| 95.333|
| 4096 (64*64)       | 95.6667| 96.6667|
| 6561 (81*81)       | 93.3333| 94.0000|

The purpose of this experiment is to evaluate the effect of feature dimensions to the accuracy and the elapsed time of the developed system. For this experiment, the feature dimension extracted by the MFCC is the manipulating variables. Three different feature dimensions are experimented i.e. 2401, 4096 and 6561 while the number of training samples and testing samples are fixed at 10 and 20, respectively.

We observed that as tabulated in Table 2, the accuracy does not depend on the feature dimension sizes. The highest accuracy of classification is achieved for both SRC (95.6667%) and KSRC (96.6667%) is at the dimension of 4096. It can also be observed that KSRC always has a higher accuracy compared to SRC due to the effect of kernel tricks applied to SRC.

On the other hand, it is obvious that the larger the feature dimension, the longer the time taken to complete the algorithm. This is because as the feature dimension increases, the dimension of the sparse code also increase thus more time is needed for both classifiers to compute the sparse solution.

For SRC, the computation time increases from 10.999s to 43.543s and subsequently to 135.210s when feature dimension increases from 2401 to 4096 and 4096 to 6561, respectively. The same trend
is observed for the KSRC. The computation time increases from 13.251s to 46.646s and subsequently to 137.284s as the feature dimension increases from 2401 to 4096 and 4096 to 6561, respectively.

For the purpose of GUI, the KSRC with 20 training samples and 4096 feature dimension are selected for system development. This configuration is used to ensure that the classification process is at its highest performance in the sense of accuracy, as proven in experiment 1 and 2.

4.3 Graphical User Interface of Frog Species Identifier
The graphic user interface (GUI) of Frog Species Identifier is programmed using MATLAB Guide. The purpose of the GUI is for end users such as wildlife or biologist researchers to identify species of frogs through their calls. Besides, it can also be as an educational software for biological teaching.

The GUI is split into two sections. The first section is ‘Select Features’ which allows user to browse and load desired frog call features into the user interface. Once the desired feature is loaded, the pathname of the data will be shown in the GUI as well. Apart from that, the push button ‘Start Identification’ plays a role to start the classifying process.

The second section of the GUI is ‘Results’. This section is updated once classification procedures ended. Once the computation of classifier is done, the common name of the frog, scientific name and sample image of the frog will be shown. Besides, a short description about the identified frog species is displayed in the GUI as well. Furthermore, there is a unique function of this GUI that allows user to listen to the identified frog call by hitting the ‘PLAY’ button in the second section. Figure 2 shows the GUI in progress while Figure 3 shows the Frog Species Identifier after the identification completed.

Figure 2. GUI: identification in progress
Figure 3. Graphical User Interface of Frog Species Identifier

References

[1] Nguyen H, Maclagan S J, Nguyen T D, Nguyen T, Flemons P, Andrews K, Ritchie E G and Phung D 2017 Proc. Int. Conf. on Data Science and Advanced Analytics (Tokyo) (IEEE) p 40

[2] Pereira F, Carneiro J and Amorim A 2008 Recent Patents on DNA & Gene Sequences 2 p 187

[3] Li Y, Huang H and Wu Z 2016 Chinese Journal of Electronics 28 p 667

[4] RubénGonzález-Hernández F, PastorSánchez-Fernández L, SergioSuárez-Guerra and AlejandroSánchez-Pérez L 2017 Applied Acoustics 119 p 17

[5] Ntalampiras S 2018 Ecological Informatics 44 p 76

[6] Welsh H H and Ollivier L M 1998 Ecological Applications p 1118

[7] Xie J, Towsey M, Zhu M, Zhang J and Roe P 2017 Ecological Indicators 82 p 13

[8] Loh C L Y Ting Yuan and Ramli D A 2013 Lecture Notes of the Institute for Computer Sciences, Social Informatics & Telecommunication vol 109 (SPRINGER) p 41

[9] Yang J, Wang J and Huang T 2011 Proc. Int. Conf. on Multimedia and Expo (Barcelona) (IEEE)

[10] Wright J, Yang A Y, Ganesh A, Sastry S S and Ma Y 2009 IEEE Transactions on Pattern Analysis and Machine Intelligence 31 p 210

[11] Zhang Z, Xu Y, Yang J, Li X and Zhang D 2015 IEEE Access 3 p 490

[12] Wright J, Ma Y, Mairal J, Sapiro G, Huang T S and Yan S 2010 Proc. IEEE 98 p 1031

[13] Ramli D A and Chien T W 2017 Procedia Computer Science p 494

Acknowledgments

This work was sponsored and supported by Universiti Sains Malaysia under Research University Grant (RU) 1001.PELECT.9014057.