Intelligent frog species identification on android operating system

W. C. Tan, H. Jaafar, D. A. Ramli, B. A. Rosdi and S. Shahrudin

Abstract— In this paper an Intelligent Frog Species Identification System (IFSIS) which works as a sensor is developed. It is designed to assist the non-experts to recognize frog species according to frog bioacoustics signals for environmental monitoring. IFSIS consists of Android devices and a server. Android device is used to record frog call signal and to display the details of the detected frog species once the identification is processed by the server. Meanwhile, feature extraction and identification process of the frog call signal are done on Intel atom board which works as server. The Mel Frequency Cepstrum Coefficient (MFCC) is used as feature extraction technique while the classifier employed is Support Vector Machine (SVM). Experimental results show that the performances of 95.33% has been achieved which proves that IFSIS can be a viable automated tool for recognizing frog species.

Keywords— Android device, Mel Frequency Cepstrum Coefficient, Support Vector Machine, Frog call.

I. Introduction

FROGS are the most common group of amphibians. Many ecologists suggest that amphibians, such as frogs, are good biological indicators because of the health of frogs is signifying the health of the whole ecosystem [1,2,3]. This is due to three reasons. Firstly, frogs require suitable habitat for both the terrestrial and aquatic environment. Secondly, frogs are in the intermediate positions of the food chains. The third reason is the skin of frogs is permeable which can easily absorb toxic and pollutants [3].

These phenomena which can be observed in our surroundings can become signs to the environmental disturbances. As reported in [4,5]. Frogs have survived for the past 250 million years in countless ice ages, asteroid crashes and other environmental disturbances but yet, one-third of these amphibian species are on the verge of extinction nowadays. So, this should be served as an alarm call to humans that if drastically wrong in our environment. Hence, smart environment monitoring is needed so as to preserve the world from frog species elimination.

Apart from for environmental monitoring, another important factor which encourages this research is owing to the discovery of the secreted peptide on frog's skin. Since, numerous bacteria are now able to develop resistance against formerly drug or antibiotic which can cause a serious threat to public health, the scientists have rekindled their interest in other alternatives for new antimicrobial agents. This new peptides have evolved a chemical resources for body protection and oxidant scavenging activities[6].

A group of Russian researchers have discovered that over 76 different antimicrobial peptides on the skin of the European Common Brown Frog (Rana Temporaria) and these peptides have potencial in preventing both pathogenic and antibiotic resistance [7]. The Alkaloid Epibatidine was also found on the skin of an Ecuadorian poison frog and it is proved to work as a powerful painkiller [8,9,10]. Finally, as frog eggs and oocytes are also involved in the cloning and embryology research so this activity will also be benefited by this proposed sensor.

Since frogs bring many advantages to ecosystem and some certain species are important for medical researches, an automatic recognition of frog species is needed. Currently, detecting and localizing certain frog species is commonly done manually by experts who is capable in recognizing the morphological characteristic of the frog. In this process, frogs or portion of the frogs need to be localized and then captured. This requires only experts with sufficient experiences and intuition to conduct the procedure. Nevertheless, the numbers of the qualified expert in this field is very limited. Furthermore, this procedure also involves intensive field sampling which is troublesome to be done manually using human visual sense [11, 12].

This work was supported in part by Universiti Sains Malaysia Short Term Grant 60311048, Research University Grant 814161 and Research University-Post Graduate Grant Scheme 8046019.

W.C. Tan, is with Intelligent Biometric Group, School of Electrical and Electronic, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Pulau Pinang, Malaysia (e-mail: twc0317@gmail.com).

H. Jaafar, is with Intelligent Biometric Group, School of Electrical and Electronic, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Pulau Pinang, Malaysia (e-mail: haryati_jaafar@yahoo.com).

D.A. Ramli is with Intelligent Biometric Group, School of Electrical and Electronic, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Pulau Pinang, Malaysia (corresponding author to provide phone: +604-5996028; e-mail: dzati@usm.my).

B.A. Rosdi is with the Intelligent Biometric Group, School of Electrical and Electronic, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Pulau Pinang, Malaysia (e-mail: eebakhtiar@usm.my).

S. Shahrudin is with the School of Pharmaceutical Sciences, USM,11800 Pulau Pinang, Malaysia (email: shahriza18@usm.my)

Due to almost all animals generate sounds either for communication or as a by-product of their living activities, so in this study, an automatic detection of frog species based on frog call is investigated. Besides, we often hear animal sound or vocalization rather than see the animal in the forest. As animals generate sounds to communicate with members of the same species, thus their vocalizations have evolved to be species-specific. Hence, recognizing animal species based on their vocalization is more effective.

Current developed frog species identification system is based on the architecture of audio biometric identification system [13,14]. Typically, this system architecture is divided into five modules which are frog call signal acquisition, signal pre-processing, feature extraction and pattern matching using classifier as illustrated in Fig. 1. This current developed frog species identification system is only implemented in computer or laptop. Due to frog species identification are always taken place in outdoor such as forest, river-side, rural area, and wetlands [15] then, the system which built with laptop are not really feasible and user-friendly. As a result, it is imperative to improve the current frog identification system to become portable and more practical. Thus, more samples can be collected and the field work can be done efficiently, conveniently, and more cost-effective. In this study, an automated system based on Intel Atom board and hand-held device android smartphone is proposed. This system is a client-server based system, where the Android smartphone acts as client and Intel Atom board acts as server. Another similar approach involved system monitoring and detection has also been researched; but for outdoor plant detection which can be found in [16]. Then, the use of smartphone platform for human behavior cognition and sensing context was reported in [17].

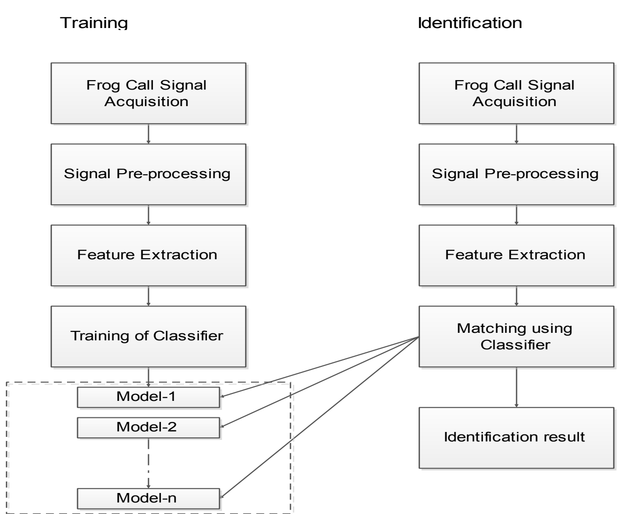


Fig. 1. Frog species identification system.

The first objective of this study is to develop data acquisition module on an Android device. Subsequently, the second objective is to develop feature extraction and classification algorithm based on Mel-Frequency Cepstrum Coefficient (MFCC) technique and Support Vector Machine (SVM) on the Intel atom board. Finally, the last objective is to set up client-server communication between Android device and Intel Atom hence to integrate the whole system as real time frog call identification which can work as frog species identification sensor.

II. Methodology

Overview of the proposed implementation of the Intelligent Frog Species Identification System (IFSIS) is shown as in Fig. 2. The overall system is described as follows:

- Input stage - Audio file of frog call is obtained either recorded by using microphone of the Android device or loaded offline from its memory cell. This audio file is then sent to the server which is an atom processor via Hypertext Transfer Protocol (HTTP).
- Processing stage - A Hypertext Preprocessor (PHP) script on the server invokes the server-side application to initiate the identification process. The recorded frog call audio are processed so as to extract the features. Subsequently, the extracted features are fed to the intelligent classifier for the identification of the types of species.
- Output stage - The identification result, which is the frog species that has been identified based on the input sound wave will be generated by the server. This result is then sent to the Android device for displaying purpose.

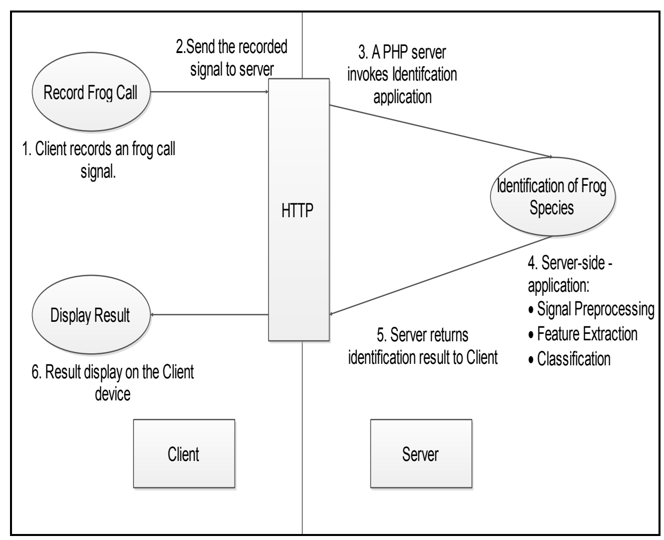


Fig. 2. Overview of IFSIS

A. Data acquisitions

Two sources of frog call samples for data acquisition are database collected by IBG Research Group, PPKEE, USM and recorded samples using Android-powered device (Samsung Galaxy S3).

The frog call samples obtained from IBG Research Group were recorded at Sungai Sedim, in Kulim, Kedah, Malaysia from 8.00 pm to 12.00 pm using Sony Stereo IC Recorder ICD-AX412F supported with Sony electret condenser microphone. The sounds samples were recorded in wav files at a sampling frequency of 44.1 kHz and are then converted to 16-bit mono. The recording dataset include samples of 15 species where the scientific name, common name and images are tabulated as in Table 1 [18,19]. The recordings were later analyzed by Praat software with the following parameters i.e call durations, average calls and standard deviations of frog calls. It was observe, depends on the species, the number of calls varies from as low as 61 and high as 148 where the average of their calls are 0.25 to 1.2s as shows in Fig. 3.

Table 1. List of frog call samples used in the project

Image, scientific name and common name		
Hylarana glandulosa Rough sided frog 	*Polypedates leucomystax* Common tree frog 	*Microhyla heymonsi* Taiwan rice frog 
Phrynoidis aspera River toad 	*Kaloula baleata* Flower pot toad 	*Fejervarya limnocharis* Grass frog 
Kaloula pulchra Asian painted bullfrog 	*Philautus mjobergi* Bubble-nest frog 	*Hylarana labialis* White-lipped frog 
Odorrana hosii Poisonous rock frog 	*Duttaphrynus melanostictus* Black-spectacled toad 	*Genus ansonia* Stream toad 
Philautus petersi Kerangas bush frog 	*Microhyla butleri* Painted chorus frog 	*Rhacophorus appendiculatus* Frilled tree frog 

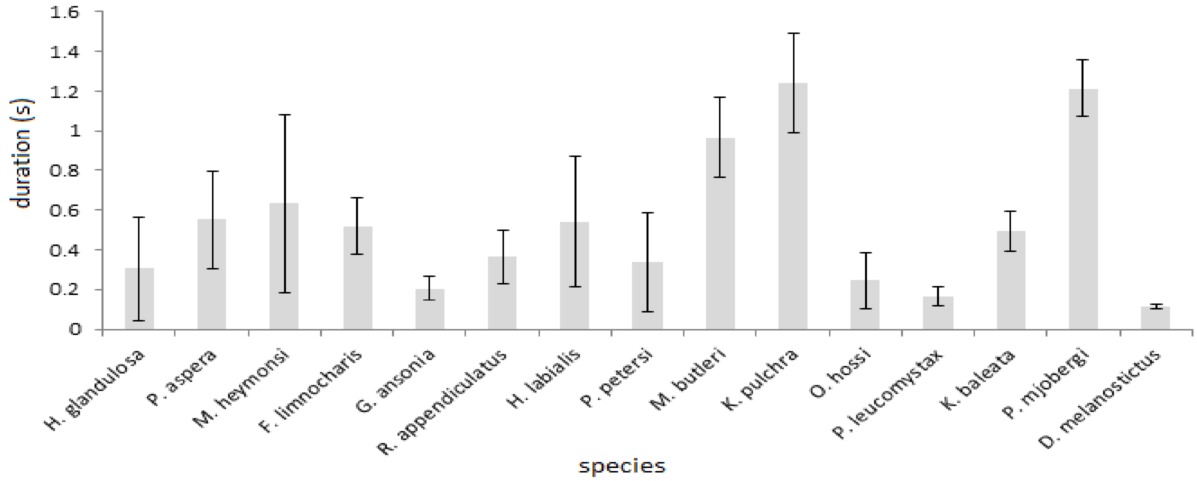
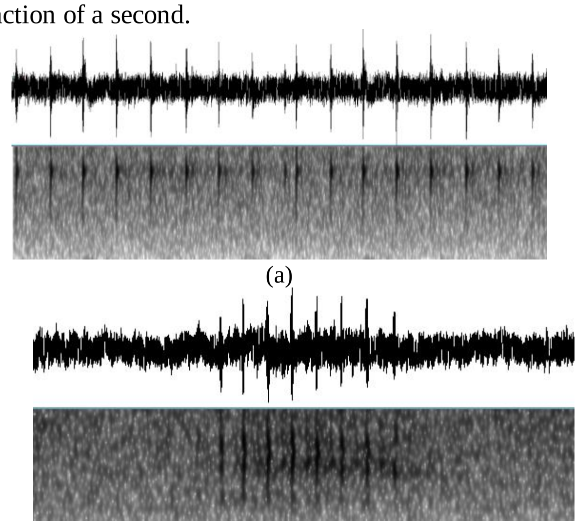


Fig. 3. Average and standard deviations for call duration of frog calls

Fig. 4 shows an example of calls waveform and spectrogram from *Microhyla heymonsi* and *Microhyla butleri*. From the figure, the waveforms from each call looked similar. However, each species has different calls based on on how the individual frog permanently changes its calls. The changing of calls is occur in a wide range of frequencies and some are long, lasting several seconds, while others last only in fraction of a second.



(a)

(b)

Fig. 4. Waveform and spectrogram for (a) *Microhyla heymonsi* (b) *Microhyla butleri.*

The frog call samples obtained from IBG Research Group is then played using speaker. The sound from the speaker is recorded using Samsung Galaxy S3. These recorded sounds are used for testing purpose for system evaluation.

B. Development of Client System in Android Device

The softwares required to develop the Android application are Eclipse IDE, Android SDK and Unified Modeling Language (UML). Eclipse IDE and Android SDK are the Android Developer Tool (ADT) which is used to develop the IFSIS Android Application for the Android client device. Unified Modeling Language (UML) is used to design the Android application for the system. A very user-friendly graphical user interface (GUI) is then designed for the application. The use case diagram which is used to determine the requirement of the specifications of the Android application is illustrated in Fig. 5.

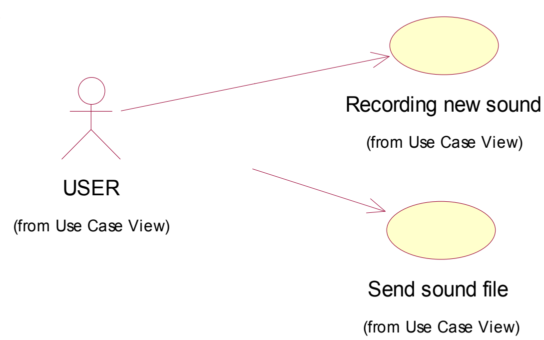


Fig. 5. Use case diagram of Android application.

IFSIS Android application is a graphical user interface application which can be run on most of the Android device. This application is used to record frog calls signal, save the signal into audio file in WAV format, upload the audio to the server, and download result from the server. The overall flowchart of this application is shown in Fig. 6.This application consists of two major parts i.e. the recording and uploading of the audio file. The Android multimedia framework includes support for capturing and encoding a variety of common audio formats. In order to perform the audio capturing or recording, some variables need to be declared and initialized. Subsequently, *Android MediaRecorder* instance is created in the next step. Next, the system needs to set the audio source, output file format, output file name, and audio encoder according to our requirements. After that, users start recording and stop recording by calling functions. Before the process ends, the system releases the *MediaRecorder* instance in order to clear the memory. The recorded sound signal is saved in WAV format on the Android device memory.

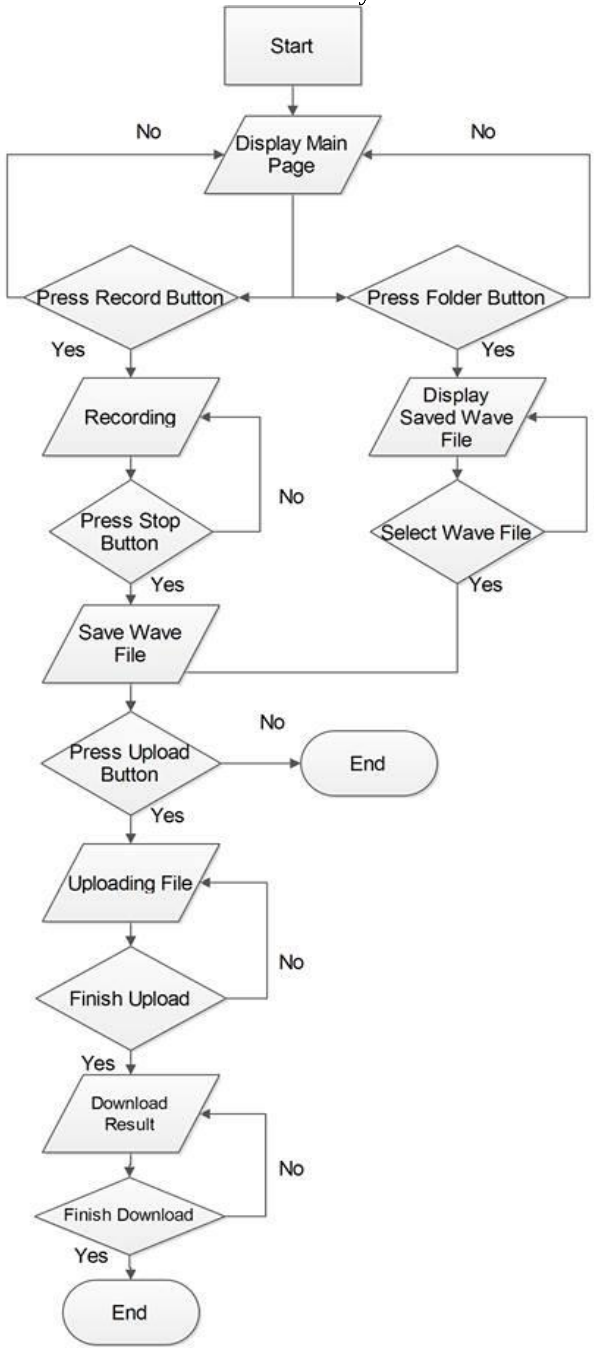
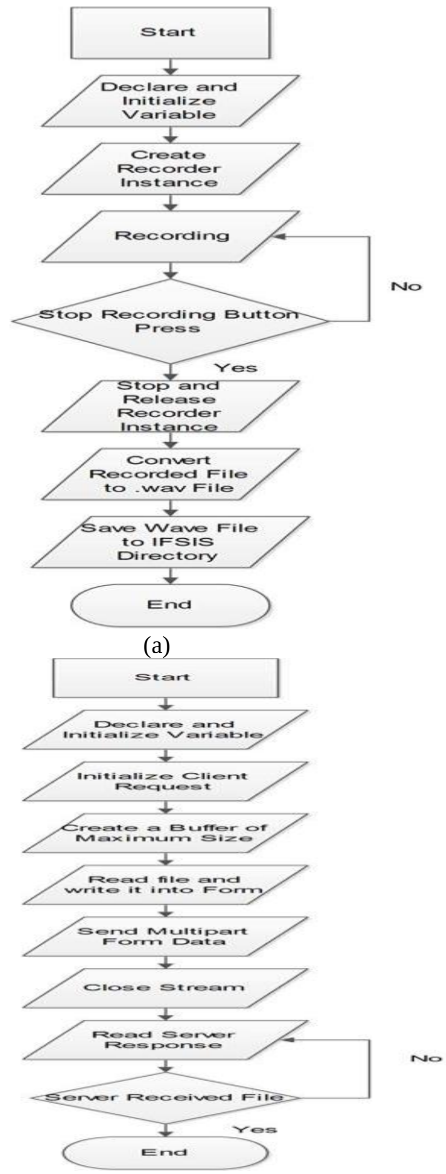


Fig. 6. Overall flowchart of IFSIS Android application

The flowchart for recording part is shown in Figure 7a.

After the frog call is recorded, it can be uploaded to the server for identification process. The application will use Hypertext Transfer Protocol (HTTP) to send and receive data. Android includes two HTTP clients: HttpURLConnection and Apache HttpClient; both of them support HTTP configurable timeouts, IPv6, and connection pooling. For IFSIS android application, HttpURLConnection is used. Next, the system creates a buffer of maximum size which is enough for the audio file to be uploaded. After that, the system will read the file and write it into form. If necessary, multipart form data is sent, and close the file input streams. While the file is being sent to the server, the application will continuously read the response from the server. The upload process is complete after the server received the file. The flowchart of uploading file is shown in Fig. 7b.



(b)

Fig. 7. Flowchart of (a) recording function and (b) uploading function developed for Android application

C. Development of Server System in Intel Atom Board

i) Server system requirement

The main hardware of the server is Intel Atom Innovation Kit 3 with some connected peripherals, such as hard disk, liquid crystal display (LCD), mouse, keyboard and wireless modem. The hard disk is used to store the operating system, files and software needed in the project. LCD monitor screen, mouse and keyboard are used to set up the identification system in the server. A wireless modem router is connected to the Gigabit Ethernet port of the server using LAN cable. Once the connection between modem and server is established, the server is ready to receive files from client devices via wireless network and send identification result back to the client. Fig. 8 shows the hardware architecture of the server part in this project. The software required is XAMPP 1.8.1 with PHP 5.4.7, Scilab 5.4.1 and Matlab 2011b. XAMPP is a free and open source cross-platform web server package. It is used to receive requests from clients and return the requested content to them. PHP is the languages used for the web server development. In this project, a PHP script is written to facilitate the communication between client and server. This script is used to receive audio files from clients, invokes Scilab script, and sends results back to the clients. Scilab is an open source, cross-platform numerical computational package with a high-level programming language. The processing of sound signal data and species identification is implemented using Scilab and Matlab.

ii) Frog call signal processing and species identification

Signal Pre-Processing

Once the audio file is received by IFSIS server, the server system will read the sound signal and execute signal pre-processing process on the signal. Signal pre-processing process in this system includes noise reduction, syllable segmentation, signal pre-emphasis, framing and windowing. These steps are employed in order to reduce computing time and increase the accuracy of the identification system.

Noise Reduction Using Band-Pass Filter.

Although it is impossible to remove all noises from the recorded sound signal, it can be minimized to certain acceptable level. The recorded frog call signals which are normally corrupted by various types of noise can reduce the accuracy of identification. In reality, noise is not merely from the environment but it can be due to the residual electronic noise signal. This electronic noise gives rise to acoustic noise heard as ‘hiss’ and it is high in frequency. Therefore, low-pass filter is used in this project so as to reduce the noise in the recorded sound signal. The cut-off frequency of the low pass filter is set lower than the noise frequencies so that the interested bandwidth can be preserved while the ‘hiss’ noise is filtered. In this project, Scilab predefined function ‘filter’ and ‘zpbutt’ are used to perform low pass filter on the recorded sound signal. Besides, Matlab predefined function ‘filter’ and ‘butter’ are used for the same purpose. The steps to develop low-pass filter are described as follows:

1. Obtain zeros and poles by using ‘zpbutt’ function in Scilab or ‘butter’ function in Matlab. This function computes the poles of a Butterworth filter of order n and cutoff frequency, f_c.
2. Obtain low-passed signal by using ‘filter’ function. The filter is set up using the zeros and poles computed in step 1.

Syllable segmentation

A syllable is a sound that a frog produces with a single blow of air from its lungs. Compared to human, frog syllables seem to be slightly less complex than human due to no-vowel-consonant and less intricate grammar [20]. In the past work, it has been indicated that zero-crossing rate (ZCR) and short-time energy (STE) are the two most important time domain and low level features which play major role in end point detection and syllable segmentation of speech [21,22]. In this project, ZCR is used together with STE for syllable segmentation. The steps for syllable segmentation using STE and ZCR are as follows:

1. The low-pass filtered signal is blocked into small frames of 20 milliseconds. The filtered signal waveform consists of a long sequence of sampled values. Thus, it is useful to break the long sequence into small frames which are quasi-stationary. The more sample points in a frame it has the less stationary it is. Therefore, a 20 millisecond frame size is chosen to compromise between sufficient sample points for accurate analysis and the quasi-stationary assumption. For a short-term sound signal (the n^{th} frame sound after framing and windowing) is as shown in (1)
$$x_n(m) = x(m)w(n-m), \qquad n-N+1 \le m \le n \tag{1}$$
Where $w()$ is window function and n is the sample that the analysis window is centered on, and N is the window size.
2. The STE of each frames is computed using (2)
$$E_n = \sum_{m=n-N+1}^{n} [x(m)w(n-m)]^2 \tag{2}$$
Where $x(m), m = 1, \ldots, N$ is the audio samples of the n^{th} frame. This simple feature can be used for detecting silent part in audio signals.
3. ZCR is a simple measure of the frequency content of a signal, especially true for narrowband signals such as sinusoids. The ZCR of each frames is computed using (3):
$$Z_n = \frac{1}{2}\sum_{m=0}^{N-1} |x(m) - x(m+1)| \tag{3}$$
Z_n is especially helpful for detecting speech from noisy background or begin and end point detection,
4. Mean and standard deviation of E_n and Z_n for the first 100 millisecond of signals are computed. It is assumed that no voiced part in this interval.
5. The maximum value of E_n from all of the frames is determined.
6. E_n thresholds are computed based on results of steps 4 and 5. This thresholds are upper threshold (ETU) and lower threshold (ETL) computed by taking some percentage of the peaks over the entire interval. Threshold for zero crossings (ZCT) based on zero crossing distribution for unvoiced speech is computed.

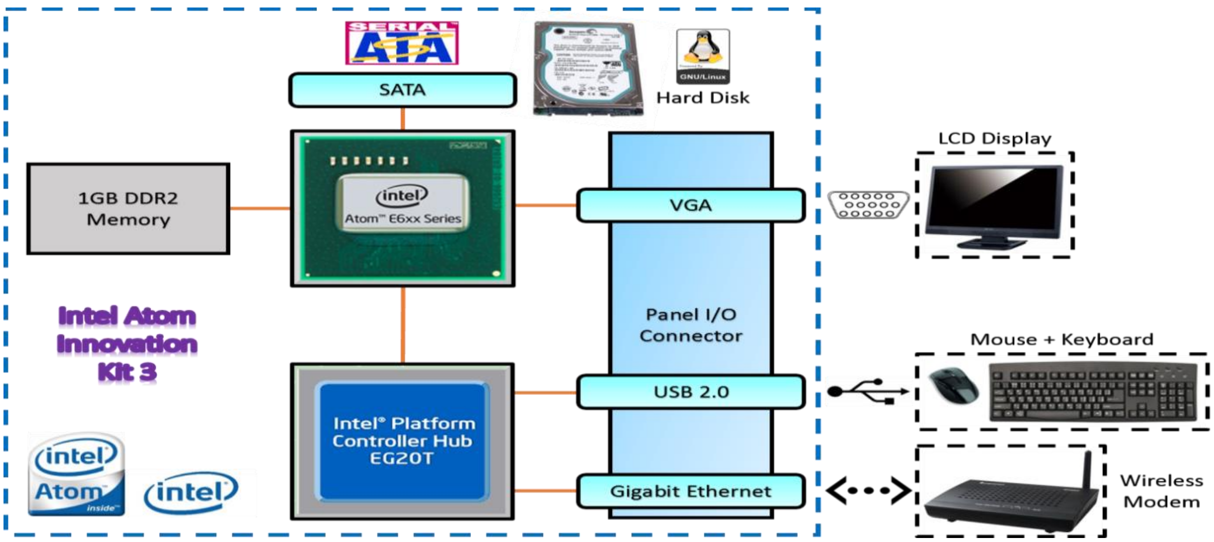


Fig. 8. Overall diagram of the server-side devices.

7. E_n thresholds are computed based on results of steps 4 and 5. This thresholds are upper threshold (ETU) and lower threshold (ETL) computed by taking some percentage of the peaks over the entire interval. Threshold for zero crossings (ZCT) based on zero crossing distribution for unvoiced speech is computed.
8. The frame with E_n exceeds the ETU threshold is determined. Find a putative starting point (N1) where E_n crosses ETL from below and a putative ending point (N2) where E_n crosses ETL from above.
9. Move backwards from N1 by comparing Z_n to ZCT, and find the first point where Z_n smaller than ZCT; similarly move forward from N2 by comparing Z_n to ZCT and finding last point where Z_n smaller than ZCT.
10. The first point and last point in step 8 are the starting point and ending point of a single syllable of voiced part signal.

Signal Pre-emphasis

The segmented syllable is first pre-emphasized to compensate the high-frequency part that was suppressed during the call production mechanism of frogs. It can also amplify the high-frequency syllables of frog call to obtain similar amplitude for all syllables. This is important because high-syllables have smaller amplitude relative to low-frequency syllables. Signal pre-emphasis is also applied to prevent numerical instability. Pre-emphasis of the segmented syllable frog call signal is implemented by filtering it with a first order FIR filter. The transfer function of this filter is in z-domain as follow:

$$H(z) = 1 - \alpha z^{-1} \qquad 0 \le \alpha \le 1 \tag{4}$$

α being the pre-emphasis parameter. Essentially, pre-emphasis filter is a first order high-pass filter in time domain.

The relationship between pre-emphasized signal and input signal is shown as follow:

$$x'(n) = x(n) - \alpha x(n-1) \tag{5}$$

A typical value for α is 0.95. This value of α gives rise to a more than 20 dB amplification of the high frequency spectrum.

Framing and Windowing

A frame-based analysis is essential for speech signals. This short-term processing is performed by framing and windowing methods. The pre-emphasized signal $x'(n)$ is framed and windowed into succession windowed sequences $x_t(n), t = 1,2, \dots, T$, known as frames. This frame can be processed individually as:

$$x'_t(n) \equiv x'(n - t \cdot Q), \qquad 0 \le t \le T, 0 \le n \le N \tag{6}$$

$$x_t(n) \equiv w(n) \cdot x'_t(n) \tag{7}$$

N is the number of samples in a frame and $w(n)$ is the impulse response of the window. Each frame is shifted by a temporal length Q given Q is smaller than N. The number of samples overlapped of one frame to the previous frame is equal to $N - Q$. This means that a total of $N - Q$ samples at the beginning of a particular frame $x_{t+1}(n)$ are duplicated from the end of the previous frame $x_t(n)$.

In this project, Q and N are set in order to overlap the frames in 50%. These frames must be in quasi-stationary so that the digitized sound signal can be represented by frames. After framing, the processing step is followed by windowing each individual frame to minimize the signal discontinuities at the beginning and end of each frame. The concept of using this step is to use the window to taper the signal to 0 at the beginning and end of each frame. This is very important because discontinuity at the begin point and last point of a frame will introduce undesirable effects in the frequency response. Frequency response of the signal is computed in feature extraction methods. Effectively, the signal is cross-multiplied by a window function as follows:

$$x'_t(n) = x_t(n) w(n), \qquad 0 \le n \le N-1 \tag{8}$$

There are several types of window functions such as rectangle, Hanning, Hamming, Blackman and Kaiser. Hamming window is a typical window applied most frequently. This window has the form,

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1 \tag{9}$$

Hamming window exhibits lower side lobes and wider main lobe than other windows. This characteristic of Hamming window reduces the leakage effect and resolution of sound signal. Thus, Hamming window is a good choice in speech recognition, because leakage will give negative effect on the signal and a high resolution is not required [23].

Feature Extraction Using MFCC

The extraction of important parametric representation of frog call signals is a crucial task in order to achieve better identification and recognition performance. Mel Frequency Cepstral Coefficients (MFCC) is one of the most commonly used feature extraction method in speech recognition. MFCC takes human hearing perception sensitivity with respect to frequencies into consideration.

After the frog call sound signal is pre-emphasized, framed, and windowed, MFCC is used to extract meaningful parameter in the frog call sound signal. The steps to implement MFCC in this project are as follows:

1. Discrete Fourier Transform (DFT) of each frame is computed for each frame. Each frame of N samples is converted from time domain into frequency domain in this step. The Fourier Transform is to convert the convolution of the glottal pulse and the vocal tract impulse response in the time domain. The DFT of all frames of the pre-processed frog call signal is:

$$X_t(\omega) = \sum_{n=1}^{N} x'_t(n) e^{-\frac{j2\pi kn}{N}}, \qquad 1 \le k \le K \tag{10}$$

   Where n is the number of samples in a frame, and k is the domain index of the DFT.

2. The signal spectrum is then processed by mel filter bank processing. The frequencies range in signal spectrum is very wide and voice signal does not follow the linear scale. Therefore, the magnitude of frequency is multiplied by Mel filter bank. This is to obtain the log energy of each triangular band-pass filter in the filter bank. The filter bank used in this project is consists of 24 triangular band-pass filter that is emphasize on processing the spectrum which frequency is below 1 kHz.

   The positions of these filters are equally spaced along the Mel frequency scale and related by following equation:

$$f_{mel} = 2595\log_{10}\left(1 + \frac{f}{700}\right) \tag{11}$$

   Where f_{mel} is the subjective pitch in Mels corresponding to a frequency in Hz. Psychophysical studies have shown that human perception of the sound frequency contents for speech signals does not follow a linear scale. Therefore, Mel scale is used to measure the subjective pitch of each tone with an actual frequency, *f*, measured in Hz.

3. Normally, log energy is obtained by computing the logarithm of square magnitude of the coefficients $Y_t(m)$. $Y_t(m)$is the m^{th} filter bank output. In this project, the log energy is obtained by computing logarithm of the magnitude of the coefficients. This is done for reducing the complexity of computing.

4. Inverse DFT is computed on the logarithm of the magnitude of the filter bank output as shown following:

$$y_t^{(m)}(k) = \sum_{m=1}^{M} \log\{|Y_t(m)|\} \cdot \cos\left(\frac{k\left(m-\frac{1}{2}\right)\pi}{m}\right), \quad k = 0, \dots\dots, L \tag{12}$$

   Where M is the number of triangular filters in the Mel filter bank, and L is the number for mel-scale cepstral coefficients. The obtained features are referred to as mel-scale cepstral coefficients, or MFCC. In this project, the value of N is 20 and L is 12. The energy within a frame is added to the 12 number of mel-scale cepstral coefficients.

5. Delta cepstrum—the first and second order time derivatives of 13 number of features which are the frame energy and mel-scale cepstral coefficients is computed.

$$y_t = \left\{y_t^{(m)}(k), e_t, \Delta\left\{y_t^{(m)}(k)\right\}, \Delta\{e_t\}, \Delta^2\left\{y_t^{(m)}(k)\right\}, \Delta^2\{e_t\}\right\} \tag{13}$$

   According to .(13), the results from first and second derivatives are added as new features. Hence, a 39-dimentional MFCC features per frames is extracted from the digitized frog call sound signal.Each feature set consists of 12 mel cepstrum coefficient, one log energy and 13 first delta cepstrum and 13 second delta cepstrum.

Identification Using SVM Classifier

A user inputs frog call signal into server system using Android client during identification process. This frog call signal is pre-processed and it meaningful parameters are extracted. These parameters or features are used to generate a testing frog call template. Then, the testing template is categorized into one species or others based on the score of the testing template with the trained model. The process of computing similarity of two different features is known as feature matching. In this project, feature matching process is carried out by Support Vector Machine (SVM) [24].

Similar to other classifiers, SVM requires training data to build model which will be used as reference to predict a new set of data into one category or the others. In this project, there are 15 species of frogs need to be identified and recognized. Thus, multi-class SVM method is used to accomplish this task. There are several ways to construct SVM classifiers for more than two classes such as one-against-all, one-against-one, and DAGSVM methods. The

SVM multi-class classification implemented in the project is the one-against-all method. In this method, 15 SVM models is constructed as 15 species of frog are going to be identified. The i^{th} SVM is trained with all of the samples in the i^{th} class with positive labels, and negative labels for the rest of the samples.
SVM assign or predict the class of x by using the following decision function:

$$class\ of\ x \equiv argmax_{i=1,\dots,k}((\omega^i)^T \phi(x) + b^i) \quad (14)$$

The largest value of the i^{th}decision function indicates that x is in i^{th} class.

In this project, 'libsvm_svmtrain' and 'libsvm_svmpredict' function in Scilab are used for SVM model training and identification, respectively. On the other hand, Matlab predefined function 'svmtrain' and 'svmpredict' are used for the same purpose.
The steps to build SVM model are listed as follows:

1. 20 samples of training data per each of the frog species is collected. This total up to 300 samples of frog call signal is used for training purpose.
2. Feature extraction process is then executed on the training samples.
3. The extracted features are then resized to 4096 feature point. This is done to ease the model training processes.
4. The i^{th} SVM model is built or trained with all of the samples in the i^{th} class with positive labels, and negative labels for samples in all other classes. The function 'libsvm_svmtrain' with polynomial kernel is used to train the SVM model.

The steps to predict or identify frog species using SVM are listed as follows:

1. A new sample of frog call data is sampled.
2. This sample is the testing data which will then proceed with feature extraction.
3. The extracted features are then resized to 4096 feature point. This is done to ease the identification processes.
4. 'libsvm_svmpredict' function is used to determine the matching rate for the testing data based on the trained SVM model.

D. Client-Server Communication using PHP

A Hypertext Preprocessor (PHP) script is written to facilitate client-server communication between Android device and Intel Atom board. This script allows a client to upload a recorded audio and returns the corresponding result from identification process. After the server is set up, the script will stand by and wait for receiving file from Android client. Once the file is successfully received, the script will invoke a Scilab script hence the identification process starts. The result from the identification process will send back to the client by the script. The flowchart of the PHP script is shown in Figure 7.

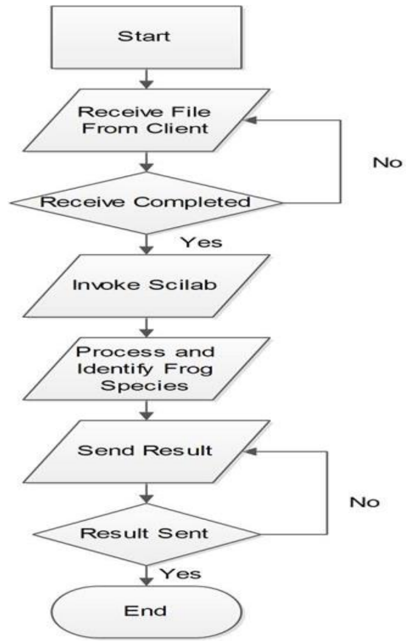


Fig. 9. Flowchart of PHP script which facilitates the client-server communication.

The steps of web server configuration are described as follows:

1. To start the server, the server machine i.e. Intel Atom Innovation Kit 3 is booted up and connected to a wireless router using a LAN cable.
2. In this project, XAMPP is used as web server application. It is one of the most robust, and offering cross platform. As the operating system Intel Atom board 3 is Fedora Linux, the Linux version of XAMPP is downloaded and installed.
3. After installation, SELinux in Fedora Linux operating system need to be deactivated. Only super user of the Linux Machine able to deactivate the SELinux. To log in as super user, the following command is used in the terminal:
   $\rightarrow su$
   Next, the system would request for super-user's password in the terminal. After successfully log in as super user, SELinux in Fedora Linux operating sytem is deactivated by using the following command line in a terminal:
   $\rightarrow setenforce\ 0$
4. The web server only can be started once the SELinux is deactivated, by calling the following command line in the terminal.
   $\rightarrow /opt/lamp/lamp\ start$
5. The written PHP script named 'v1.php' is then stored in the server folder path as follows:
   $\rightarrow /opt/lampp/htdocs/vup1$
6. After step 5 is completed, the server is now ready to communicate with the client.

Fig. 10 illustrates the overall structure of the developed IFSIS.

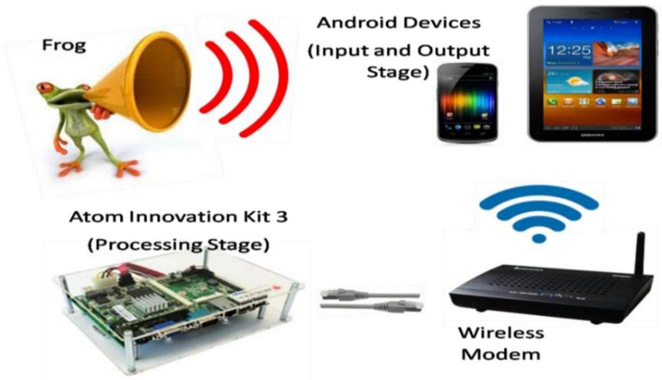


Fig. 10. Overall Structure of IFSIS

III. RESULTS AND DISCUSSION

In this experiment, 30 samples of frog call syllables are evaluated for each species. Training data which consists of 10 samples from each species are used to build the SVM model, while 20 samples from each species are randomly selected to test the system. Based on the results in Table 2, Scilab-IFSIS achieves 95.33% of accuracy which is considered high and this reaches the expected accuracy which has been set to 90%. The accuracy of Matlab-IFSIS is 95.67% which is slightly higher than the accuracy of Scilab-IFSIS.

Confusion matrix in Table 2 and 3 shows the true positives and false positives of the Scilab-IFSIS system on frog calls samples. It can be observed from the confusion matrix that eight samples are tabulated under 'unknown' column. These samples of frog call are unidentified by the system. Besides that, there are six false positives are made by the system which are three from Philautus Mjobergi, and each from Hylarana Labialis, Genus Ansonia, and Microhyla Butleri. Out of 300 samples of frog call, a total number of 286 samples are correctly identified by the system based on the number of true positives.

A. Performances of IFSIS in term of processing time

The processing time of the identification processes are also recorded to evaluate the efficiency of IFSIS. In order to calculate the processing time, 15 frog call samples (1 sample for each species) with duration of 15 seconds were taken as the testing samples to record the processing time. The processing time is defined as the time taken to upload the frog call audio file samples by Android client to server until the result of the identification is displayed on the client. The difference of the processing time for each identification process is caused by the length of frog call syllables. Each species of frog exhibits unique syllable trend which is also different in length. The longer syllable of the frog call, the longer time is taken for the identification process. The results are tabulated in Table 4.

As observed from the above table, the processing time using Matlab-IFSIS is 24.00 sec. This performance is better compared to Scilab-IFSIS which is 27.17 sec in average. This system is considered efficient if the processing time is short and this achieves the expectation of the project. The expected processing time was set to 60 seconds or less.

Table 2. True positive and false positive of Matlab-IFSIS

		Predicted class																ACC (%)
	sp	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Unknown	
Actual Class	1	20																100
	2		18														2	85
	3			20														100
	4				20													100
	5					20												100
	6						18										2	90
	7						3	15									2	75
	8								20									100
	9									20								100
	10										19						1	95
	11											19			1			95
	12												19			1		95
	13													20				100
	14														19	1		95
	15															20		100
Mean Recognition Accuracy																		95.33

Sp1: HylaranaGlandulosa
Sp2: PhrynoidisAspera
Sp3: MicrohylaHeymonsi
Sp4: FejervaryaLimnocharis
Sp5: Genus Ansonia
Sp6: RhacophorusAppendiculatus
Sp7: HylaranaLabialis
Sp8: PhilautusPetersi
Sp9: MicrohylaButleri
Sp10: KaloulaPulchra
Sp11: OdorranaHosii
Sp12: PolypedatesLeucomystax
Sp13: KaloulaBaleata
Sp14: PhilautusMjobergi
Sp15 DuttaphrynusMelanostictus

Table 3. True positive and false positive of Scilab-IFSIS

		Predicted class																ACC (%)
	sp	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Unknown	
Actual Class	1	20																100
	2		17														3	85
	3			20														100
	4				20													100
	5					20												100
	6						18										2	90
	7						3	15									2	75
	8								20									100
	9									20								100
	10										19						1	95
	11											19			1			95
	12												19			1		95
	13													20				100
	14														19	1		95
	15															20		100
Mean Recognition Accuracy																		95.33

Sp1: HylaranaGlandulosa Sp6: : RhacophorusAppendiculatus Sp11: OdorranaHosii
Sp2: PhrynoidisAspera Sp7: HylaranaLabialis Sp12: PolypedatesLeucomystax
Sp3: MicrohylaHeymonsi Sp8: PhilautusPetersi Sp13: KaloulaBaleata
Sp4: FejervaryaLimnocharis Sp9: MicrohylaButleri Sp14: PhilautusMjobergi
Sp5: Genus Ansonia Sp10: KaloulaPulchra Sp15 DuttaphrynusMelanostictus

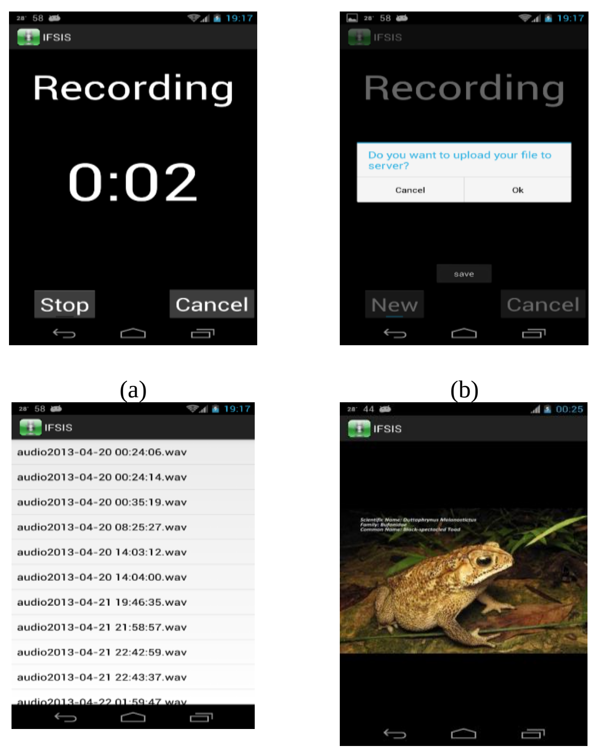
Table 4. Processing time using Scilab-IFSIS and Matlab-IFSIS

Scientific name	Processing time (second)	
	Matlab-IFSIS	Scilab-IFSIS
HylaranaGlandulosa	20.34	23.02
PhrynoidisAspera	26.07	30.80
MicrohylaHeymonsi	21.33	24.12
FejervaryaLimnocharis	22.77	25.26
Genus Ansonia	22.97	25.19
RhacophorusAppendiculatus	24.21	27.85
HylaranaLabialis	25.83	29.66
PhilautusPetersi	22.99	25.19
MicrohylaButleri	23.05	26.33
KaloulaPulchra	27.80	31.58
OdorranaHosii	25.12	28.83
PolypedatesLeucomystax	25.61	29.13
KaloulaBaleata	23.04	25.71
PhilautusMjobergi	28.82	32.48
DuttaphrynusMelanostictus	20.06	22.44
Average time (second)	24.00	27.17

As a conclusion, in term of accuracy and processing time of the IFSIS, Matlab is slightly powerful than Scilab. However, Scilab-IFSIS is still a better option for this project in order to minimize the development cost for the whole system.

The android application GUI is shown as shown in Fig. 11. It consists of three main layout which facilitates user to perform the following procedures:

- Recording the frog call signal.
- Uploading the frog call audio file to Atom board (server).
- Displaying the result on the android device.



(a) (b)

(c) (d)

Fig. 11. User interface of the Android application for (a) recording sound signal, (b) alert dialog after users press "Stop" button, (c) audio file directory; and (d) result

IV. Conclusion

Intelligent Frog Species Identification system (IFSIS) has achieved high accuracy in identifying frog species correctly and efficiently. The identification of frog species which can be done using smartphone interface takes advantages of the android-device's graphic capabilities to make the system easy to use and portable. The system GUI assists user to record frog call signal, upload to-be-recognized frog call audio file to server, and read the identification result while the processing can be done via server on atom processor. From the experimental results, it shows that the proposed system is promising and can work as remote sensing for frog species identification. Thus, with the innovative, practical, reliable and affordable IFSIS, we may now suggest that *"Now, everybody can be a physiological research expert in identifying frog species."*

Acknowledgment

The authors would like to thank the financial support provided by Universiti Sains Malaysia Short Term Grant 60311048, Research University Grant 814161 and Research University-Post Graduate Grant Scheme 8046019 for this project.

References

[1] P. Kathryn, "Where have all the frogs and toads gone," *Bioscience,* vol. 40, no. 6, pp. 2-4, 1996.

[2] T.J.C. Beebee, R.A. Griffiths, "The amphibian decline crisis: a watershed for conservation biology?," *Biological Conservation*, vol. 125, no.3, pp. 271-285, 2005.

[3] C. Carey, "Kathryn Phillips: 1995, Tracking the Vanishing Frogs, Penguin Books," *Climatic Change*, vol. 37, no. 3, pp. 565-567. 1997.

[4] N.H. Shubin, F. A. Jenkins, "An Early Jurassic jumping frog," *Nature* vol. 377, no. 6544, pp.49-52, 1995.

[5] S.N. Stuart, J.S. Chanson, N.A. Cox, B.E. Young, A.S.L. Rodrigues, D.L. Fischman, R.W. Waller, "Status and Trends of Amphibian Declines and Extinctions Worldwide," *Science*, vol. 306, no. 5702, pp. 1783-1786, 2004.

[6] X. Liu, R. Liu, L. Wei, H. Yang, K. Zhang, J. Liu, and R. Lai, "Two novel antimicrobial peptides from skin secretions of the frog, *Rana nigrovittata*," *Journal of Peptide Science*, vol. 17, pp. 68-73, 2010.

[7] T.Y. Samgina, E.A. Vorontsov, V.A. Gorshkov, E. Hakalehto, O. Hanninen, R.A. Zubarev, A.T. Lebedev, "Composition and Antimicrobial Activity of the Skin Peptidome of Russian Brown Frog Rana temporaria," *Journal of Proteome Research*, vol. 11, no. 12, pp. 6213-6222, 2012.

[8] A. Gomes, B. Giri, A. Saha, R. Mishra, S.C. Dasgupta, A. Debnath, A. Gomes, "Bioactive molecules from amphibian skin: their biological activities with reference to therapeutic potentials for possible drug development," *Indian journal of experimental biology*, vol. 45. No. 7, pp. 579-593, 2007.

[9] C. Qian, T. Li, T.Y. Shen, L. Libertine-Garahan, J. Eckman, T. Biftu, S. Ip, "Epibatidine is a nicotinic analgesic," *European journal of pharmacology*, vol. 250,no. 3, pp.R13-R14, 1993.

[10] S.C. Clayton, A.C. Regan, "A total synthesis of (±)-epibatidine," *Tetrahedron letters*, vol. 34, no. 46, pp. 7493-7496, 1993.

[11] G.G. Yen, Q. Fu, "Automatic frog calls monitoring system: a machine learning approach," *International Journal of Computational Intelligence and Applications*, vol. 1, no. 02, pp.165-186, 2001.

[12] A. Taylor, G. Watson, G. Gordon, H.M. Callum, "Monitoring frog communities: an application of machine learning," Proceedings of Eighth Innovative Applications of Artificial Intelligence Conference, Portland Oregon, 1996.

[13] C.J. Huang, Y.J. Yang, D.X. Yang, Y.J. Chen, "Frog classification using machine learning techniques," *Expert Systems with Applications* vol. 36, no. 2, pp. 3737-3743, 2009.

[14] H. Jaafar, and D. A. Ramli, "Automatic syllables segmentation for frog identification system," presented at the 2013 9th IEEE Int. Colloquium on Signal Processing and its App, 90-95.

[15] A. Jansen, M. Healey, "Frog communities and wetland condition: relationships with grazing by domestic livestock along an Australian floodplain river," *Biological Conservation*, vol. 109, no. 2, pp. 207-219, 2003.

[16] V. Dworak, J. Selbeck, K.H. Dammer, M. Hoffmann, A. Zarezadeh, C. Bobda, "Strategy for the Development of a Smart NDVI Camera System for Outdoor Plant Detection and Agricultural Embedded Systems," *Sensors*, vol. 13, no. 2, pp. 1523-1538, 2013.

[17] L. Pei, R. Guinness, R. Chen, J. Liu, H. Kuusniemi, Y. Chen, L. Chen, J. Kaistinen, "Human Behavior Cognition Using Smartphone Sensors," *Sensors*, vol. 13, no. 2, pp. 1402-1424, 2013.

[18] S. Shahrudin, J. Ibrahim, M.S. Anuar, "The Amphibian Fauna of Lata Bukit Hijau, Kedah, Malaysia," *Russian Journal of Herpetology*, vol. 18, no. 3, pp. 221-227, 2011.

[19] D.R. Frost, "Amphibian Species of the world: an Online Reference," Version 5.5 (31 January 2011). American Museum of Natural History, New York, 2011.

[20] D. Stowell, M.D. Plumbley, "Birdsong and C4DM: A survey of UK birdsong and machine recognition for music researchers," Tech. Rep. C4DM-TR-09-12, Queen Mary University of London, 2010.

[21] C. Panagiotakis, G. Tziritas, "A speech/music discriminator based on RMS and zero-crossings," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 155-166, 2005.

[22] A. Ghosal, R. Chakraborty, S. Haty, B.C. Dhara, S.K. Saha, "Speech/music classification using occurrence pattern of zcr and ste," Third International Symposium on Intelligent Information Technology Application, 2009.

[23] J.W. Picone, "Signal modeling techniques in speech recognition," *Proceedings of the IEEE*, vol. 81, no. 9, pp. 1215-1247, 1993.

[24] V. Vapnik, *The nature of statistical learning theory*, Springer-Verlag, New-York, 1995.