

Article

Implementation of Artificial Intelligence for Classification of Frogs in Bioacoustics

Kuo-Wei Chao ¹, Nian-Ze Hu ^{2,3,*}, Yi-Chu Chao ⁴, Chin-Kai Su ⁴ and Wei-Hang Chiu ⁴

¹ Department of Mechanical Engineering, National Cheng Kung University, Tainan 701, Taiwan; e14076148@ncku.edu.tw

² Smart Machinery and Intelligent Manufacturing Research Center, National Formosa University, Yunlin 632, Taiwan

³ Department of Information Management, National Formosa University, Yunlin 632, Taiwan

⁴ Fudan Senior High School, Taoyuan 324, Taiwan; tracy.chao30126@gmail.com (Y.-C.C.); chin.kai0205@gmail.com (C.-K.S.); johnny770826@gmail.com (W.-H.C.)

* Correspondence: drhu@nfu.edu.tw

Received: 9 November 2019; Accepted: 24 November 2019; Published: 26 November 2019



Abstract: This research presents the implementation of artificial intelligence (AI) for classification of frogs in symmetry of the bioacoustics spectral by using the feedforward neural network approach (FNNA) and support vector machine (SVM). Recently, the symmetry concept has been applied in physics, and in mathematics to help make mathematical models tractable to achieve the best learning performance. Owing to the symmetry of the bioacoustics spectral, feature extraction can be achieved by integrating the techniques of Mel-scale frequency cepstral coefficient (MFCC) and mentioned machine learning algorithms, such as SVM, neural network, and so on. At the beginning, the raw data information for our experiment is taken from a website which collects many kinds of frog sounds. This in fact saves us collecting the raw data by using a digital signal processing technique. The generally proposed system detects bioacoustic features by using the microphone sensor to record the sounds of different frogs. The data acquisition system uses an embedded controller and a dynamic signal module for making high-accuracy measurements. With regard to bioacoustic features, they are filtered through the MFCC algorithm. As the filtering process is finished, all values from cepstrum signals are collected to form the datasets. For classification and identification of frogs, we adopt the multi-layer FNNA algorithm in machine learning and the results are compared with those obtained by the SVM method at the same time. Additionally, two optimizer functions in neural network include: scaled conjugate gradient (SCG) and gradient descent adaptive learning rate (GDA). Both optimization methods are used to evaluate the classification results from the feature datasets in model training. Also, calculation results from the general central processing unit (CPU) and Nvidia graphics processing unit (GPU) processors are evaluated and discussed. The effectiveness of the experimental system on the filtered feature datasets is classified by using the FNNA and the SVM scheme. The expected experimental results of the identification with respect to different symmetry bioacoustic features of fifteen frogs are obtained and finally distinguished.

Keywords: artificial intelligence (AI); feedforward neural network approach (FNNA); symmetry; Mel-scale frequency cepstral coefficient (MFCC); machine learning (ML); graphics processing unit (GPU); support vector machine (SVM); bioacoustics

1. Introduction

The lives on Earth are closely relevant to environmental changes on various spatial and temporal scales [1]. The success of human societies intimately relies on the living elements of natural and

managed systems. Even though geographical range limits of species are naturally changing with time-variants, climate change is gradually impelling a universal redistribution of lives on Earth [2,3]. To know more about the life distribution or migration of animals, related prospective detection and research in bioacoustics should be conducted [4,5].

There are obvious characteristics of animals that can be used as sensors for the early detection of what are most concerned. Behavior [6,7] and sound [8–10] are both such characteristics. It is probable to make use of sound to detect some phenomena, no matter which acoustic field it is [11]. If things produce sounds, there are existing ways to interpret and predict possible meanings conveyed by sound. For example, things that are in good condition sound energetic, whereas they sound obviously weak under bad conditions. As a specific example, emission with a choking sound as well as abnormal acoustic/vibration energy of automobiles might indicate some problems [12]. By the same token, different sound features may symbolize corresponding characteristics of an entity. This also explains why sounds are commonly used by animals to communicate or interact [8,13]. From practical experience, distinct animal acoustic features reveal different contents of meaningful information [14]. Furthermore, the same information conveyed by a particular sound can be transformed into other appearances without loss of its meaning. For example, even when they are under water, sounds generated by some animals are converted to a series of bubbles [15,16]. Owing to the mass acoustic communication among animals, automated acoustic monitoring can not only give an appropriate way to survey different species in their natural way of life, but can also provide a convenient and cost-effective way to monitor target species efficiently. At the same time, it can reduce the need of manual monitoring [17,18]. Nowadays, bioacoustic feature classification has become a quite useful tool for experts to collect and process data to generate useful information for monitoring the ecological system [19]. The goal is to look for clues for understanding the bioacoustic features of an animal based on the perspective of its bioacoustic mechanisms. Remotely monitoring living creatures on a real-time basis allows us to gather significant acoustic data information, which can be used not only for predicting changes of environment but also observing other relevant phenomena like global warming, animal extinction, natural disasters, and various diseases [20,21]. Research in this field has been mushrooming. According to the information released from the Biodiversity Research Center, Academia Sinica, a study called “Deeply listening to diverse creature under extreme climate” is now conducted for detecting acoustic signals and collecting data to make a training dataset model. Related equipment is shown as Figure 1.

Because animals living in nature are quite sensitive to the environment, it is natural for them to react quickly to the environment they are in [22]. This suggests the possibility to predict coming natural phenomena from observing the bioacoustic changes of animals [23,24]. Take frogs as an example: before it rains, frog will croak much louder. Also, experts have gradually begun to focus on the topic of automated identification of animal sounds [25] because animal sounds are relatively easy to recognize [14]. Therefore, we hope that bioacoustic changes could help us predict the dynamics of natural phenomena.

Recently, the world has entered an era of AI. Big data, the basis of AI, becomes a critical tool to establish an optimal decision model which generates samples from the big data for machines to predict the best solution. Datasets are used to develop machine training to make the model get familiar with the trend. Training machines to learn would keep correcting the error value toward its minimization for fitting the best situation until the model is optimized. Machines play an important role in any industrial process of the modern digital world. Various methods have been applied for assuring machine operations by collecting and analyzing many different types of useful information such as vibration, acoustic, and temperature trends [26]. In our experiment, we choose fifteen frog calls to be analyzed by our model [27]. Frog calls are easy to hear and collect, and often used to provide different useful classifications that are intended [28]. The reasons are threefold. The first reason is that the spectrogram structure of frog signals is relatively simple and can be in the form of frequency tracks or harmonic waves, making spectral features clear to recognize under a complicated environment. The second is that the basic vocal unit of frog call syllables is often short and includes different durations

in the classification [29]. Finally, automatic exploration and collection of acoustic data are highly convenient and effective, having given rise to many researches [30–32].

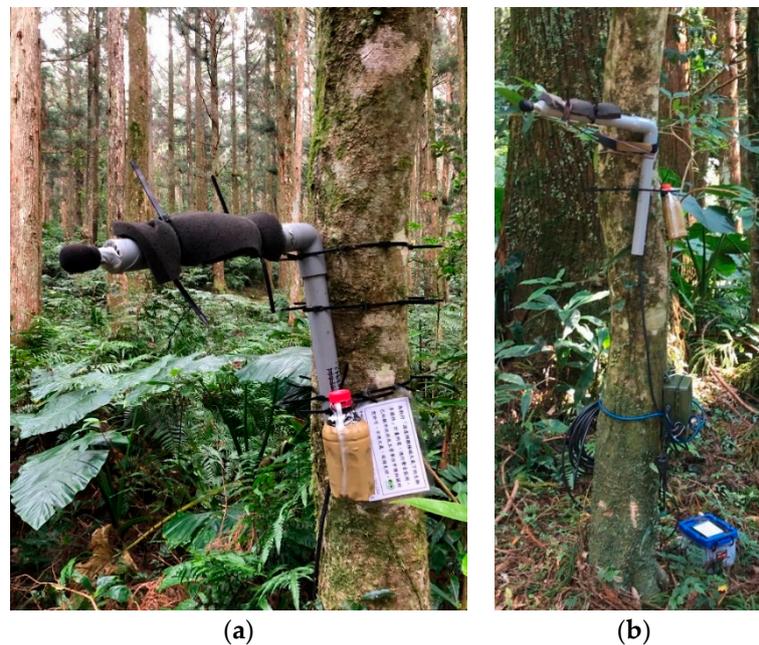


Figure 1. (a) An ultrasound microphone is equipped to record the sounds from the overall environment; (b) The data acquisition module is used for the microphone to collect real-time signals.

Machine learning is known as the brain of AI, meaning that it could deduce a mode from a large amount of data and that it is capable of analyzing and learning from the known data and then making possible predictions based on testing data [33,34]. It also provides researchers in classical statistics with extra data modelling techniques [6]. Such an automatic learning process concept has its assured place in the modern world. In case the output space has no structure except if two components from the output are equal or not, it is called the problem of classification learning. Each component of the output space is called a class. The learning algorithm solving the classification problem is called classifier. The task of such classification problems is to assign new inputs to one of a few discrete classes or categories. This problem characterizes most pattern identification tasks [35].

In this paper, we will particularly describe how computation methods of machine learning are able to contribute to problem-solving in the bioacoustic field. Since there might be the case that some parts of spectrum signals of the collected data are not robust enough, a pre-emphasis process is used to improve the quality of the signals in such a way that these signals will be set to go through a high-pass filter to enhance the frequency magnitude feature. Meanwhile, it can also filter from outside environment some signals that are considered to be useless while keeping the rest intact. Then, a well-known speech algorithm, the Mel-frequency cepstral coefficients (MFCC), is used to distinguish between speech technologies [36,37]. Using such an algorithm, original signals from frog sounds are transformed into spectrum [38]. When useful spectrum signals are all gathered to form a big dataset, the dataset can serve as a training model for the machine. If the model is stably robust, then it is appropriate to apply the model to identify and classify new datasets [39]. Accordingly, we provide some unknown datasets for two processors, central processing unit (CPU) and graphics processing unit (GPU), to perform the sound identification of the frog species and further classify the characteristics of each frog to get the result. Nevertheless, there are pros and cons between the two kinds of processor, which will be discussed.

One of the often-applied algorithms is neural networks [40]. Neural networks are able to learn patterns or relationships from training big data and generalize what they have learned,

and then they will extract expected results from the resulting data [41,42]. The approximation calculation of neural networks is done on the basis of connectionism. After going through the model training process, the network is a kind of machine that approximately leads inputs to desired outputs [43]. In our experimental process of classification, the feedforward neural network approach (FNNA) is adopted to conduct machine learning [44,45]. A feedforward neural network is a set of connected neurons, where information only flows in the forward direction, from inputs to outputs [46]. The other algorithm adopted in the experiment is support vector machine (SVM). In machine learning, the support vector machine method belongs to supervised learning models in connection with learning algorithms that analyze data and identify patterns, used for regression and classification analysis [47,48]. Additionally, MATLAB is applied in the experiment and some of its notable features such as time duration, algorithms, and efficiency will be discussed.

Relevant to the study, there have been numerous researchers investigating the information content of vocal tone. For example, the connection between body mass and frequency used in vocal performances has been edited by Wallschlager (1980). Similar results have been discussed by Boeckle et al. (2009) for 76 species of frogs where 25% of the dominant frequency has been announced by body size [49,50]. Once we are able to analyze bioacoustic features for various purposes, we can further apply the technique to the analysis of other kinds of sound. The analysis of sound can be further developed to deal with issues such as the problem of voice disorders and the detection of diseases related to diseases of the heart or the lung and so on [51–53]. Therefore, we hope that this kind of bioacoustic research could be practically applied in the biomedical field for saving lives. For example, through collecting information on one's breath, we can monitor breath features and identify breathing movements [9]. This will help provide more effective medical therapy and improve the quality of human health care. We, therefore, expect that results of this research can lay a strong foundation for these applications.

2. Theory of Bioacoustics Signal Processing

2.1. Bioacoustic Feature Extraction

In the beginning, it is necessary to start the analysis with an algorithm called Pre-emphasis filtering. The equation is defined as:

$$F_{\text{pre}}(z) = 1 - az^{-1}, \quad (1)$$

the primary reason for using this concept is that as the vocal cord vibrates, the vocal-cord side is regarded as a series of pulse signals passing through a glottal shaping filter. The output of airflow velocity waveform has a characteristic of -12 dB/oct high frequency attenuation. At the lip side, it is seen as having a radiation impedance, and the signal produces a high-pass filtering effect with a high-frequency enhancement of 6.

For a discrete-time signal of a speech, we use a fixed-length window to observe the signals inside the window and analyze them. Finally, speech features can be identified. If we choose a rectangular window function, original signals will be retained inside the window but will be set to be zero outside, making signals of both sides become discontinuous like they were cut off. Discontinuity of both sides may result in extra sound, and from the frequency domain, the voice spectrum will be destroyed. In order to avoid this, the Hamming-window function is appropriately adopted. It will let the window extract the signals and two sides could slowly decrease [54].

$$w(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2n\pi}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The bioacoustic signals of frogs must first be processed by a discrete time conversion mechanism. Then the discrete time signals are transferred to spectrum domain. The cepstrum of a signal is the inverse discrete-time Fourier transform (IDTFT) of the logarithm of the magnitude of the DTFT of the signal, which is defined as:

$$C_n = \frac{1}{2\pi} \int_0^{2\pi} \log \left(\frac{1}{\prod_{k=1}^{k=r} (1 - x_k Z^{-1})} \right) dZ$$

$$\Rightarrow C_n = -\frac{1}{2\pi} \int_0^{2\pi} \log \left(\prod_{k=1}^{k=r} (1 - x_k Z^{-1}) \right) dZ, \quad (3)$$

or

$$C[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log |X(e^{i\omega})| e^{i\omega n} d\omega, \quad (4)$$

where $X(e^{i\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-i\omega n}$ is the spectrum.

$C[n]$ is a discrete function of index n apparently. If an input sequence $x[n]$ is generated by sampling an analog signal, we can consider $x[n] = x_a(n/f_s)$ and associate index n with time in this transformation.

Suppose that a linear-filtered kind of bioacoustic signal takes the following form: $y[n] = h_d[n] \times x[n]$ rather than $y[n] = x[n]$. If the analysis window is long in comparison with the length of $h_d[n]$, the short time interval cepstrum of a frame of the filtered bioacoustic signal $y[n]$ will probably take the following form:

$$c_m^{(y)}[n] = c_m^{(x)}[n] + c^{(h_d)}[n], \quad (5)$$

where $c^{(h_d)}[n]$ will come out more or less the same in every frame. Therefore, if we can estimate the value of $c^{(h_d)}[n]$, which is assumed to be non-time-varying, we may get $c_m^{(x)}[n]$ at every frame from $c_m^{(y)}[n]$ by $c_m^{(x)}[n] = c_m^{(y)}[n] - c^{(h_d)}[n]$. Another way to remove the influence of linear distortions is applying the mechanism of cepstrum to every window because the distortions are the same in each frame. Therefore, the influence of linear distortions could be gotten rid of by an easy first difference operation:

$$\Delta c_m^{(y)}[n] = c_m^{(y)}[n] - c_{m-1}^{(y)}[n]. \quad (6)$$

It is obvious that if $c_m^{(y)}[n] = c_m^{(x)}[n] + c^{(h_d)}[n]$ and $c^{(h_d)}[n]$ is independent of m , then $\Delta c_m^{(y)}[n] = \Delta c_m^{(x)}[n]$, meaning that the linear distortion effects are disappeared.

The fact that the weighted cepstrum distance measure conveys exactly the same practical meaning as the distance measure in frequency domain is very important for models of human perception of sound and provides a basis for the frequency analysis conducted in the inner ear. Because of this, the Mel-frequency cepstrum coefficients method is born.

As just shown, a short-time Fourier analysis goes first, and the outcome is the DFT values for the m^{th} frame. Then the DFT values are gathered for each band and weighted by a triangular weighting function. The Mel-spectrum of the m^{th} frame is defined as:

$$MF_m[r] = \frac{1}{A_r} \sum_{k=L_r}^{U_r} |V_r[k] X_m[k]|^2, \quad (7)$$

where $V_r[k]$ is the weighting function for the r^{th} filter.

$$A_r = \frac{1}{A_r} \sum_{k=L_r}^{U_r} |V_r[k]|^2, \quad (8)$$

is a normalizing factor of the r^{th} Mel-filter. The normalization is needed so that a flat Mel-spectrum can be produced by an ideally flat input Fourier spectrum. For each frame, a discrete cosine transformation (DCT) of the Mel-filter output is computed to generate the function $mfcc[n]$ shown below:

$$\text{mfcc}[n] = \frac{1}{R} \sum_{r=1}^R \log(\text{MF}_m[r]) \cos\left[\frac{2\pi}{R}\left(r + \frac{1}{2}\right)n\right], \quad (9)$$

$\text{mfcc}[n]$ is evaluated for a number of coefficients N_{mfcc} .

2.2. Frogs Classification Method

2.2.1. Feedforward Neural Network Approach

Machine learning is seen as the process of using a resource-based calculation to implement some learning algorithms. Actually, machine learning is defined as a complex computation process of automatic pattern identification and intelligent decision-making embedded in the process of training sample data. A feedforward neural network approach, widely used for supervised bioacoustics classification, can be shown as Figure 2.

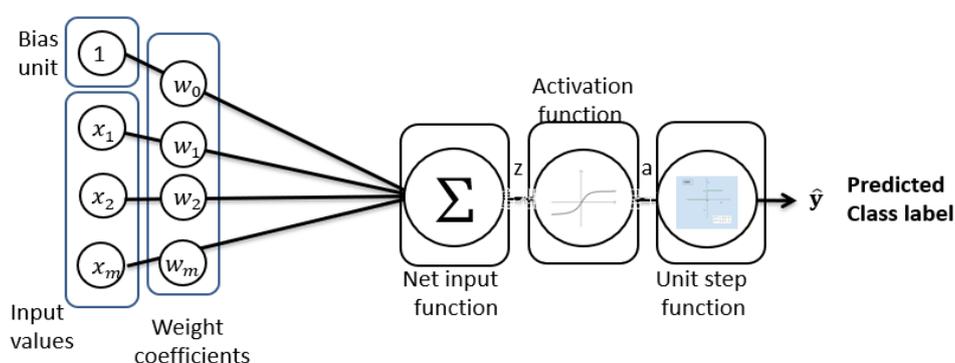


Figure 2. The structure of the feedforward neural network approach algorithm is shown. Basically, the structure interprets the general concept of neural networks, which contains layers and layers of neurons. Here, the information flows only in the forward direction from the input to the output in the feedforward neural network approach (FNNA) scheme.

Therefore, feedforward multilayer networks with a sigmoid nonlinear function are often termed multilayer perceptrons (MLP). Machine learning methods can be classified into three groups: supervised learning, unsupervised learning, and reinforcement learning. In this research, we use the FNNA structure to carry out the proposed, but only part of, supervised learning. The structure is defined [55–58] as follows:

$$z = \sum_j w_j x_j = w^T x$$

$$\hat{y} = \begin{cases} 1, & g(z) \geq 0 \\ -1, & \text{otherwise} \end{cases} \quad (10)$$

The feedforward neural network is a general machine learning method which transforms inputs into outputs that will be in line with targets. Through non-linear signal processing in a random number of connected groups of artificial neurons, the so-called hidden layers are formed. When the FNNA is used, it is important to control a set of weights to minimize the error in the process of classification. A useful instrument commonly seen in many learning ways is the least mean square (LMS) convergence criterion. The goal of feedforward neural network is narrowing the gap between the ground truth Y and the output $f(X; W)$ of the ground truth by applying $E(X) = (Y - f(X; W))^2$. The procedure of such an approach relies on both the weighting scheme and the transfer function T_f , which are essential to the connections between neurons. The general function of the feedforward neural network approach is described below:

$$y_j^1 = T_f \left(\sum_i x_i \times w_{ji}^1 \right). \tag{11}$$

2.2.2. Support Vector Machine Approach

The support vector machine combines machine learning methods and statistical methods, aiming to generate a mapping from training dataset for establishing a route between input and output. Basic concept can be expressed as Figure 3.

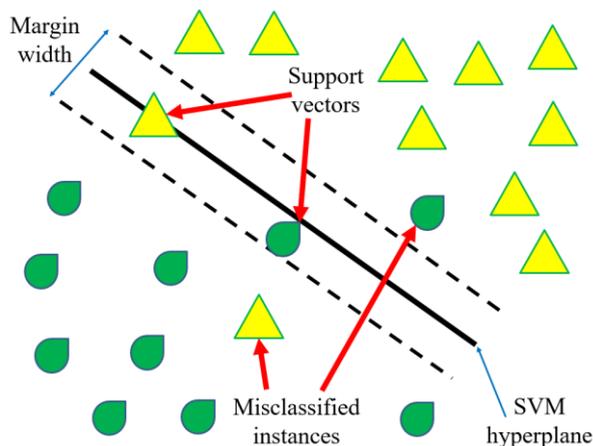


Figure 3. The structure of the linear support vector machine algorithm. Generally, this method is used to generate the input-output mapping from training dataset.

$$f(x) = w^t x + b. \tag{12}$$

It finds the hyperplane which maximizes the separating margin between the two classes [59]. This hyperplane could be found by minimizing the cost function shown below:

$$\min J(w) = \frac{1}{2} w^t w = \frac{1}{2} \|w\|^2, \tag{13}$$

subject to the following separability constraints:

$$y_i (w^t x_i + b) \geq 1; \text{ where } i = 1, 2, \dots, l. \tag{14}$$

A slack variable could be used to relax the separability constraints:

$$y_i (w^t x_i + b) \geq 1 - \varepsilon_i; \text{ where } \varepsilon_i \geq 0 \text{ and } i = 1, 2, \dots, l. \tag{15}$$

The linear support vector machine with a separating hyperplane can classify things only into two classes, which is evidently not sufficient for making medical predictions where classifications into several classes are often needed. When features of interest in the sample space could not be separated by only hyperplanes, nonlinear techniques should be called for. Therefore, the nonlinear form of such an algorithm is usually applied for complex applications in the real world.

Now consider the input vector $x \in R^d$ that is transformed into the feature vector $\Phi(x)$ through a nonlinear mapping $\Phi : R^d \rightarrow R$. Then the problem is solved by assuming a kernel function $K : R^d \times R^d \rightarrow R$ defined as:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j), \tag{16}$$

where x_i and x_j indicate any pair of input vectors.

Thus, the optimal separating contours are defined according to the function f given by:

$$f(x) = \sum_{k=1}^1 \alpha_k y_k K(x_k, x) + \beta, \tag{17}$$

where α_k and β are scalars and rely on x_k and y_k , $k = 1, \dots, 1$.

The support vector machine offers great generalization capabilities. It is robust for high dimensional data, well-suited to conduct training, and performs well in comparison with traditional artificial neural networks. However, the support vector machine is very sensitive to uncertainties. The larger the dimensions of the space, the higher the probability the support vector machine will be trapped into a lengthy learning process. For some real-time applications, the evaluation of function f may be difficult to manage. Therefore, a balance has to be maintained between the generalization properties of the SVM and its sluggishness when faced with learning from many large databases. In brief, the key is making a good choice of the related input variables from the dataset so that dimensions of the space can be reduced, and approximation functions can be efficient and accurate [60].

3. Results and Verification

The analysis is mainly carried out by applying a digital MFCC algorithm and using the acoustic sensors. In Figure 4, we establish an experimental structure of our research arrangement process.

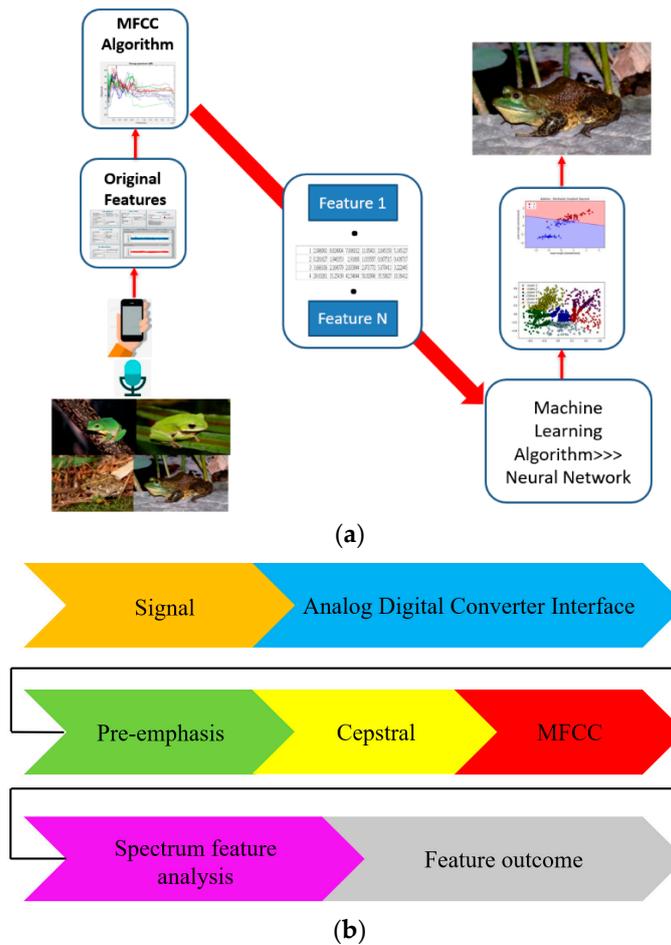
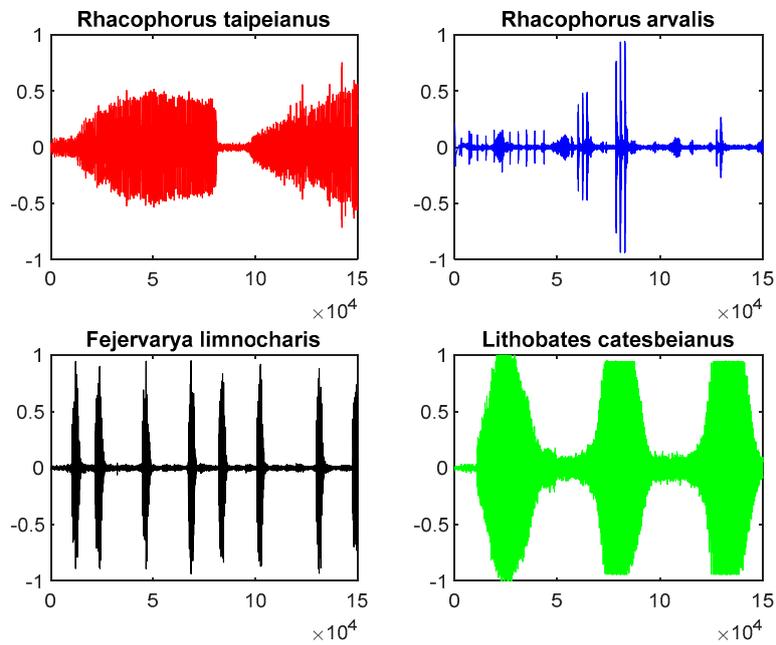
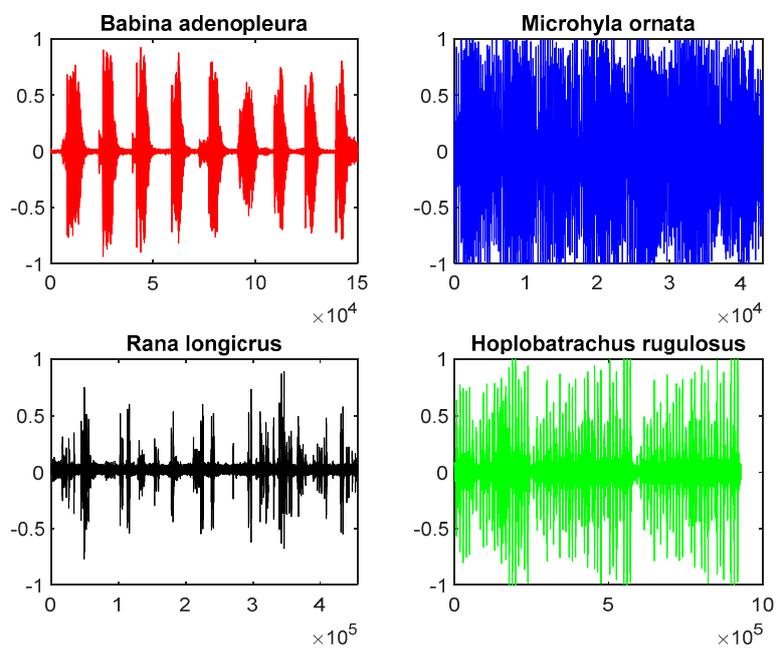


Figure 4. (a) This structure introduces the experimental process of the frog classification. Briefly, the process includes raw data collection→pre-emphasis→Mel-scale frequency cepstral coefficient (MFCC) filtering→features classification→identification results; (b) The diagram indicates the core elements of the procedure in the experiment.

Our original information on frog sound is from the digital learning website [27], and their sound information files are shown in Figure 5. Through adjusting pre-emphasis coefficient “a”, twenty-five filtered features of each frog are shown in Figure 6.



(a)



(b)

Figure 5. Cont.

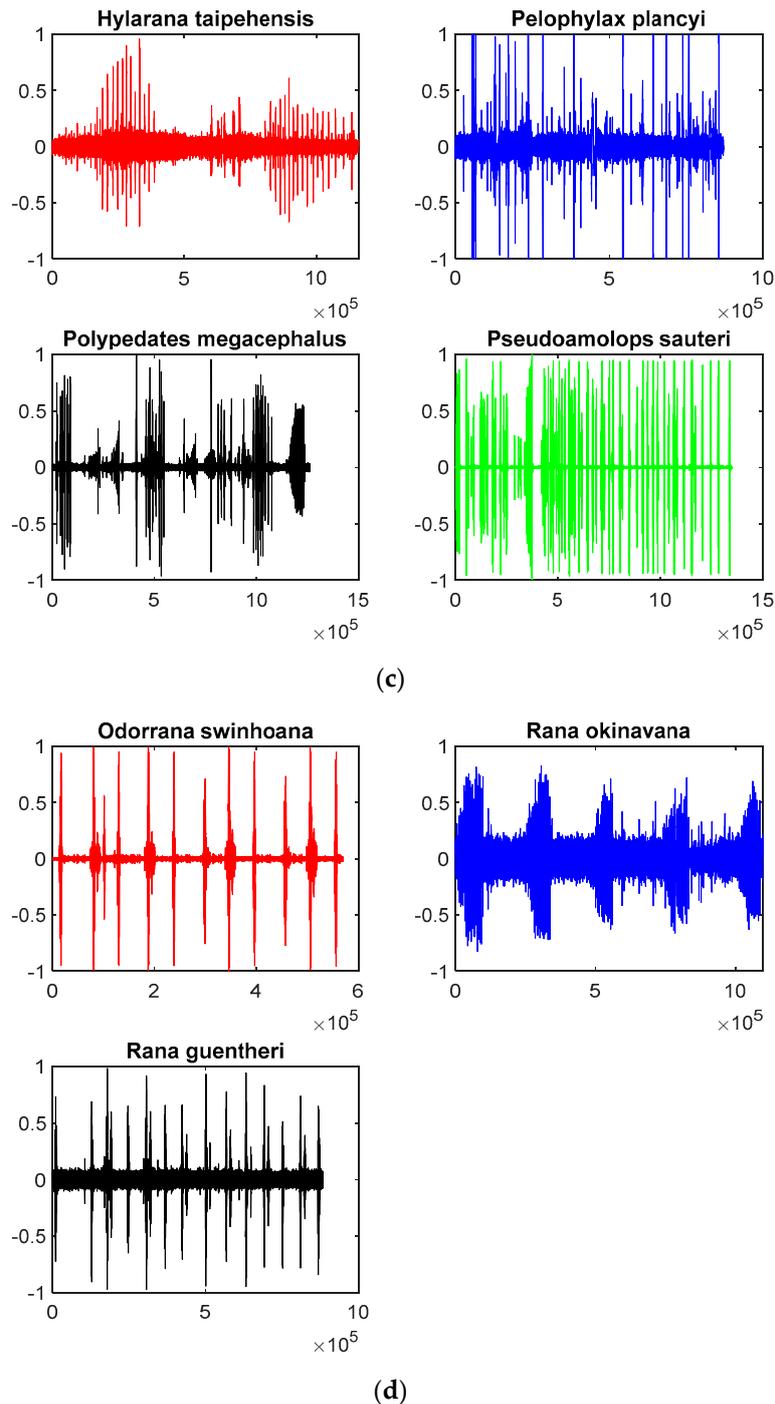
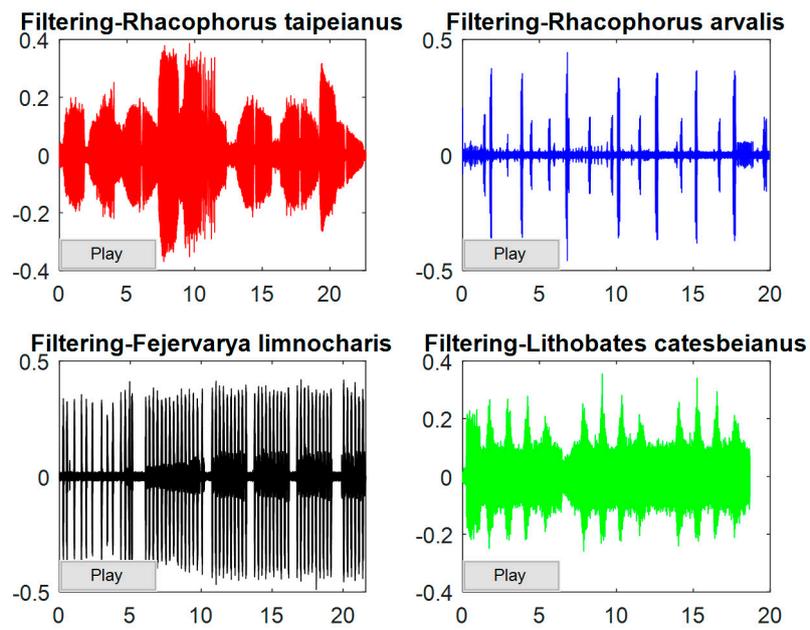
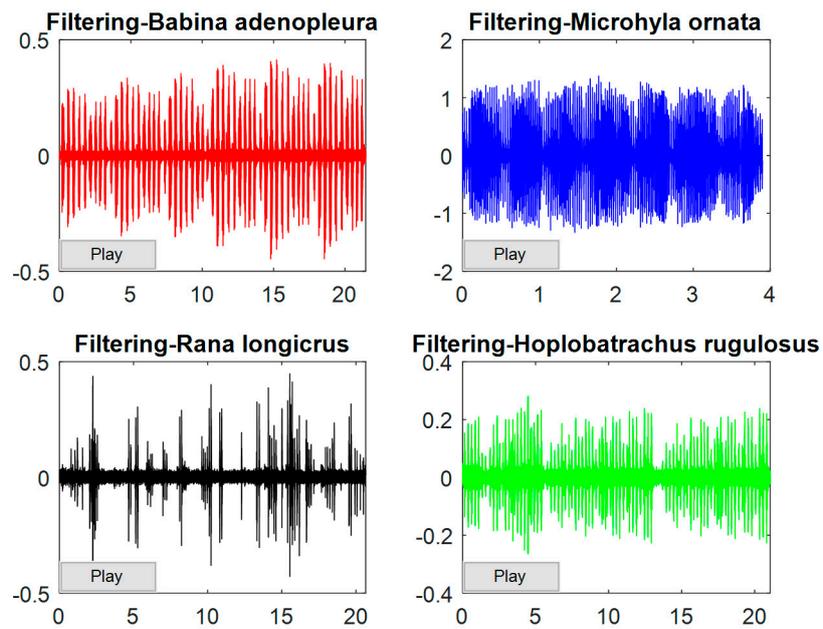


Figure 5. The raw data information on fifteen kinds of frogs is presented in four forms. The time length of each raw data is approximately 20 s. (a) The red signal indicates the raw data information of *Rhacophorus-taipeiensis*; the blue one indicates that of *Rhacophorus-arvalis*; the black one indicates that of *Fejervarya-limnocharis*; the green one indicates that of *Lithobates-catesbeianus*; (b) The red signal indicates the raw data information of *Babina-adenopleura*; the blue one indicates that of *Microhyla-ornata*; the black one indicates that of *Rana-longicrus*; the green one indicates that of *Hoplobatrachus-rugulosus*; (c) The red signal indicates the raw data information of *Hylarana-taipehensis*; the blue one indicates that of *Pelophylax-plancyi*; the black one indicates that of *Polypedates-megacephalus*; the green one indicates that of *Pseudoamolops-sauteri*; (d) The red signal indicates the raw data information of *Odorrana-swinhoana*; the blue one indicates that of *Rana-okinavana*; the black one indicates that of *Rana-guentheri*.



(a)



(b)

Figure 6. Cont.

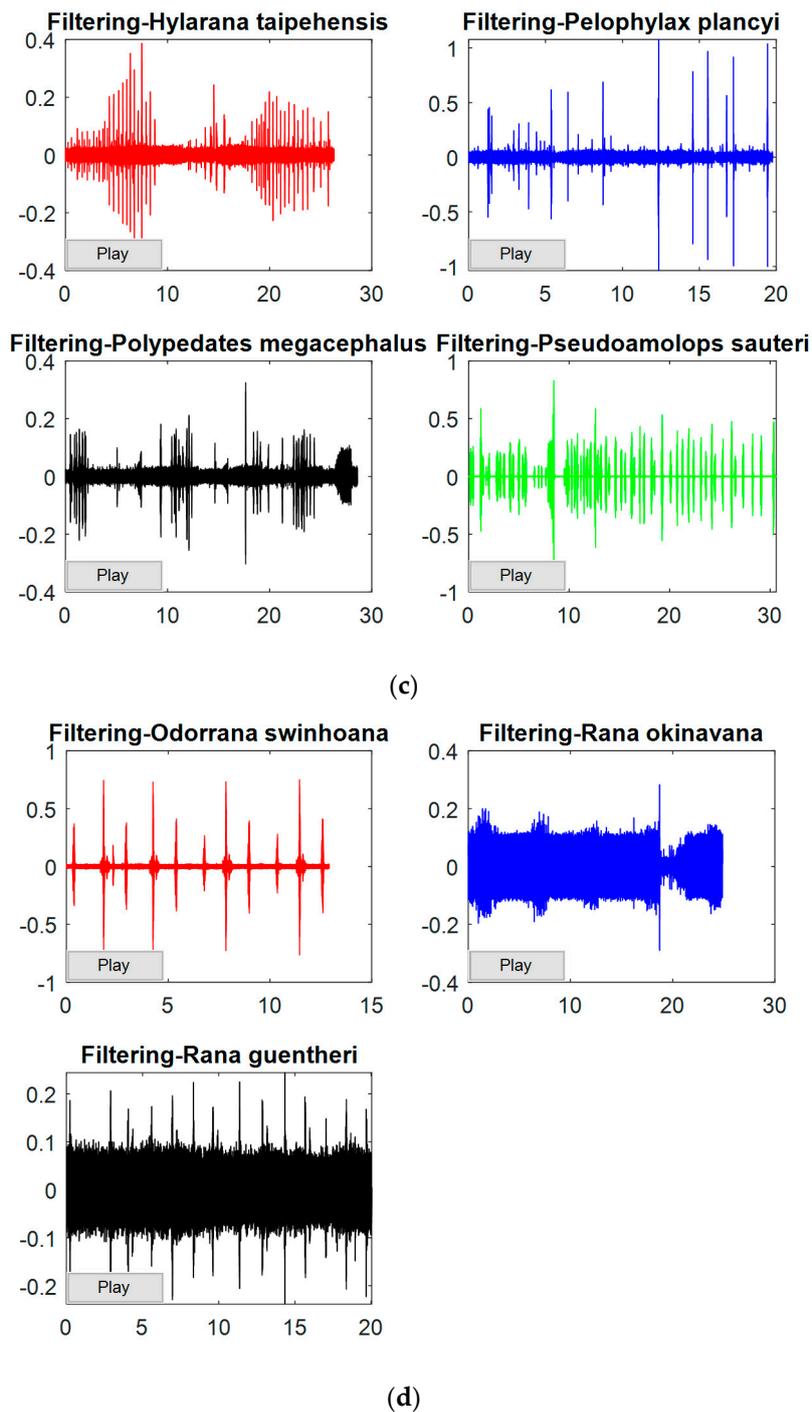


Figure 6. After a filtering process with the value of ‘a’ equal to 0.95, the raw data are pre-emphasized, and the filtering results are shown in four forms. (a) The red signal indicates the filtered data information of *Rhacophorus-taipeiensis*; the blue one indicates that of *Rhacophorus-arvalis*; the black one indicates that of *Fejervarya-limnocharis*; the green one indicates that of *Lithobates-catesbeianus*; (b) The red signal indicates the filtered data information of *Babina-adenopleura*; the blue one indicates that of *Microhyla-ornata*; the black one indicates that of *Rana-longicrus*; the green one indicates that of *Hoplobatrachus-rugulosus*; (c) The red signal indicates the raw data information of *Hylarana-taipehensis*; the blue one indicates that of *Pelophylax-plancyi*; the black one indicates that of *Polypedates-megacephalus*; the green one indicates that of *Pseudoamolops-sauteri*; (d) The red signal indicates the raw data information of *Odorrana-swinhoana*; the blue one indicates that of *Rana-okinavana*; the black one indicates that of *Rana-guentheri*.

The MFCC filtering algorithm is applied to the transformation from time-domain signal to special spectrum features as shown in Figure 7. Also, a dataset for training is formed and shown in Figure 8.

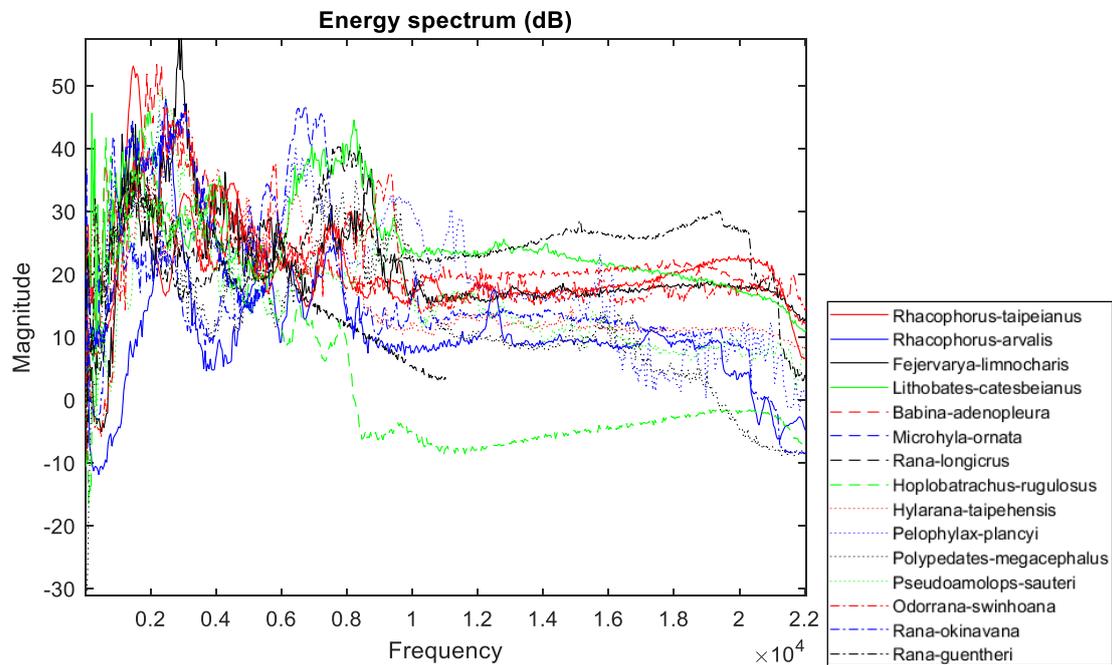


Figure 7. It is a spectrum diagram of frog bioacoustic signals through the MFCC algorithm, plotted with 60,000 points for each frog. The MFCC filtering process transforms the original signals into spectra.

1_98	1.466898	7.511003	11.10963	3.347651	7.981834	3.479017	5.028172	3.842393	3.089069	3.568865	2.641629	3.007519	2.884411	2.828157	2.831066
1_96	2.886009	14.53816	20.027	5.50636	12.33419	4.994189	6.844353	4.985911	3.842907	4.306457	3.096168	3.454463	3.249204	3.138255	3.101546
1_95	3.600004	18.09981	24.68737	6.685942	14.78383	5.884067	7.943084	5.700143	4.327273	4.7913	3.401949	3.760321	3.503408	3.357609	3.295597
1_94	4.315088	21.67242	29.39771	7.89272	17.31523	6.817197	9.1097	6.468888	4.855843	5.326524	3.74374	4.105494	3.793275	3.609931	3.520657
1_92	5.746937	28.83418	38.89565	10.34942	22.51269	8.757288	11.56369	8.106983	5.998089	6.497341	4.501806	4.879584	4.451367	4.188943	4.042427
1_90	7.179927	36.00701	48.44609	12.83596	27.80573	10.75113	14.10838	9.822854	7.208462	7.750902	5.32337	5.727057	5.180205	4.836897	4.632296
1_88	8.613495	43.18537	58.02313	15.33777	33.14866	12.77354	16.70244	11.58197	8.457774	9.052904	6.18317	6.619797	5.953861	5.529615	5.267435
1_86	10.0474	50.3669	67.61547	17.84843	38.52075	14.8128	19.326	13.36727	9.731043	10.38515	7.067264	7.54175	6.75696	6.252238	5.933324
1_85	10.76443	53.95846	72.41548	19.10598	41.21417	15.83672	20.64533	14.26668	10.37391	11.05921	7.515752	8.010544	7.166471	6.621721	6.27476
1_84	11.48151	57.55041	77.21741	20.36465	43.9113	16.86281	21.9685	15.16951	11.01997	11.73734	7.96757	8.4834	7.580147	6.995505	6.620693
1_82	12.91575	64.73525	86.82577	22.88461	49.31427	18.92006	24.62386	16.9633	12.31964	13.10331	8.879157	9.438853	8.417532	7.753484	7.323496
1_80	14.3501	71.92101	96.43863	25.40718	54.72597	20.98242	27.28832	18.80529	13.62698	14.47916	9.798876	10.40431	9.265265	8.522249	8.037685
1_78	15.78452	79.10745	106.0548	27.93165	60.14406	23.04852	29.95945	20.63331	14.93998	15.86232	10.72464	11.37722	10.12075	9.299128	8.760476
1_76	17.21899	86.29439	115.6733	30.45756	65.56695	25.11745	32.63562	22.46589	16.25727	17.25102	11.655	12.35583	10.98217	10.08224	9.489904
1_75	17.93624	89.88801	120.4834	31.72095	68.27983	26.15275	33.97522	23.38356	16.91721	17.94706	12.12157	12.84687	11.41468	10.4757	9.856647
1_74	18.6535	93.48172	125.2938	32.98458	70.99353	27.18854	35.31568	24.302	17.57788	18.64404	12.58894	13.38888	11.84823	10.87025	10.22455
1_72	20.08805	100.6693	134.9158	35.51246	76.42303	29.26135	37.9988	26.14089	18.90111	20.04047	13.52572	14.32546	12.71798	11.66215	10.96336
1_70	21.52262	107.8572	144.539	38.04104	81.85487	31.33553	40.68439	27.98202	20.22646	21.43964	14.4648	15.31489	13.59073	12.45721	11.70555
1_68	22.95721	115.0453	154.1631	40.57018	87.28859	33.41083	43.37198	29.82497	21.55353	22.84106	15.40574	16.30664	14.46591	13.25486	12.45052
1_65	25.10913	125.8277	168.6007	44.36474	95.44204	36.52545	47.40638	32.59214	23.54672	24.94655	16.81996	17.79778	15.78239	14.45524	13.57217
1_63	26.54376	133.0162	178.2266	46.89491	100.8792	38.60277	50.09761	34.4384	24.87692	26.35205	17.76431	18.79379	16.66204	15.25763	14.32223
1_60	28.69573	143.799	192.6664	50.69071	109.0368	41.71985	54.13642	37.20957	26.87392	28.46249	19.18268	20.29009	17.98395	16.46378	15.45008
1_57	30.84771	154.582	207.1071	54.48704	117.1962	44.83799	58.17714	39.98249	28.87256	30.57509	20.60285	21.78866	19.30824	17.67246	16.58065
1_53	33.71705	168.9596	226.3625	59.54948	128.0777	48.99682	63.56712	43.68186	31.53945	33.39455	22.49863	23.78951	21.07688	19.28714	18.09143
1_50	35.86907	179.743	240.8047	63.34671	136.2401	52.11672	67.61101	46.45767	33.54084	35.51073	23.9218	25.29182	22.40512	20.50002	19.22653

Figure 8. Twenty-five features of the first frog. The green area is labeled as X_YY, where X means the number order of frog and YY means the pre-emphasis coefficient ‘a’ with only two decimals shown. The orange area shows that each frog has four hundred spectrum feature values.

When preprocessing is done, the key analysis follows. The feedforward neural networks algorithm is used to train the bioacoustic model datasets of frog. The computation platform we use for conducting the analysis is “MATLAB R2019a-academic use” authorized by National Cheng Kung University, Tainan, Taiwan. And the two processors adopted for classification are GPU and CPU. For each processor, two optimizer functions, “traingda” and “trainscg”, are executed. Going from the current layers and the next layers of neurons, many comparisons and tests are conducted so that the two processors

can learn and attempt to predict an expected model. If the new testing data become the input, this algorithm structure can monitor the data information and predict the classification results by the learning program for training datasets. The calculation of machine learning contains a concept called the “Deep Feedforward Neural Network” method. Furthermore, the “back-propagation algorithm” is used to update the weighting scheme. In the deep learning computation procedure, the back-propagation algorithm will be carried out by iterations not only to minimize the loss function but to get close to the best performance of the testing model. With regard to the optimizer functions, the “traingda” (GDA) is a neural network optimizer function that resets the weighting scheme and the bias value based on gradient descent with an adaptive learning rate (LR) for an optimal change in the learning rate during the training process. This can allow the computing condition to keep stable. In short, the “traingda” function can automatically alter the LR by adapting to the convergence factor from the error values.

$$dx = lr \times dperf/dx. \quad (18)$$

Each variable is adjusted according to gradient descent. At each epoch, if performance drops toward the goal, then the LR is increased by the factor “lr_inc”. In contrast, if performance becomes larger than the factor “max_perf_inc”, the LR is revised down by the factor “lr_dec”. Also, a general fixed learning rate η is then replaced by an adaptive learning rate:

$$\frac{c_1}{[\text{number of iterations}] + c_2}. \quad (19)$$

The “trainscg” is a network optimizer function which renovates bias values and weights according to the scaled conjugate gradient (SCG) method. It trains any network as long as its net input, weight, and transfer functions have derivative functions. Backpropagation is used to calculate derivatives of performance “perf” with respect to the weights and bias vectors x .

There are three important diagrams from GPU and CPU: Regression, Performance, and Training State. In Regression, the slope (or we can say efficiency), marked as R, of the linear regression shows us the level this machine training is able to achieve. The point of Performance is to see green circle if it can catch up with where the epochs are now. Regarding the Training State, it indicates whether the efficiency of the analysis is going up or down. Basically, we try to improve the accuracy of the classification by changing the training function and processor. The rest of the set of parameters are the same. Related parameters are set up as the learning rate, 0.00008, the initial epoch, 30,000, and the hidden layers being [100, 200, 400, 200, 100] for five layers of neurons.

From Figures 9–22, the identified patterns can be categorized into five groups. Figures 9–11 report results based on processor GPU with optimizer GDA while Figures 12–14 are based on processor CPU with optimizer GDA. Figures 15–17 are from processor GPU with optimizer SCG and Figures 18–20 are from processor CPU with optimizer SCG. Also, Figures 21 and 22 report the results of the SVM. Additionally, when it comes to the characteristic of GPU, it is commonly thought of as a rapid computation compared with CPU. With regard to computation in MATLAB platform, a GPU does not need too much time to finish the identification. Summary results are shown in Table 1. Comparatively, Table 2 makes the comparison in total time between the feedforward neural network and the support vector machine.

Table 1. Comparisons of results between GPU and CPU.

Machine Learning Processor	Optimizer Function	Final Epochs	Total Time (Second)	R-Scores
GPU	traingda	12,234	9.228	0.99798
CPU	traingda	9668	577.4620	0.99818
GPU	trainscg	183	4.3264	0.99832
CPU	trainscg	191	11.5490	0.99800

Table 2. Comparisons of results between neural networks and support vector machine.

Machine Learning Algorithm	Total Time (Second)	R-Scores
Neural networks (conducted by GPU with training function SCG)	4.3264	0.99832
Support vector machine	5.1480	1.00000

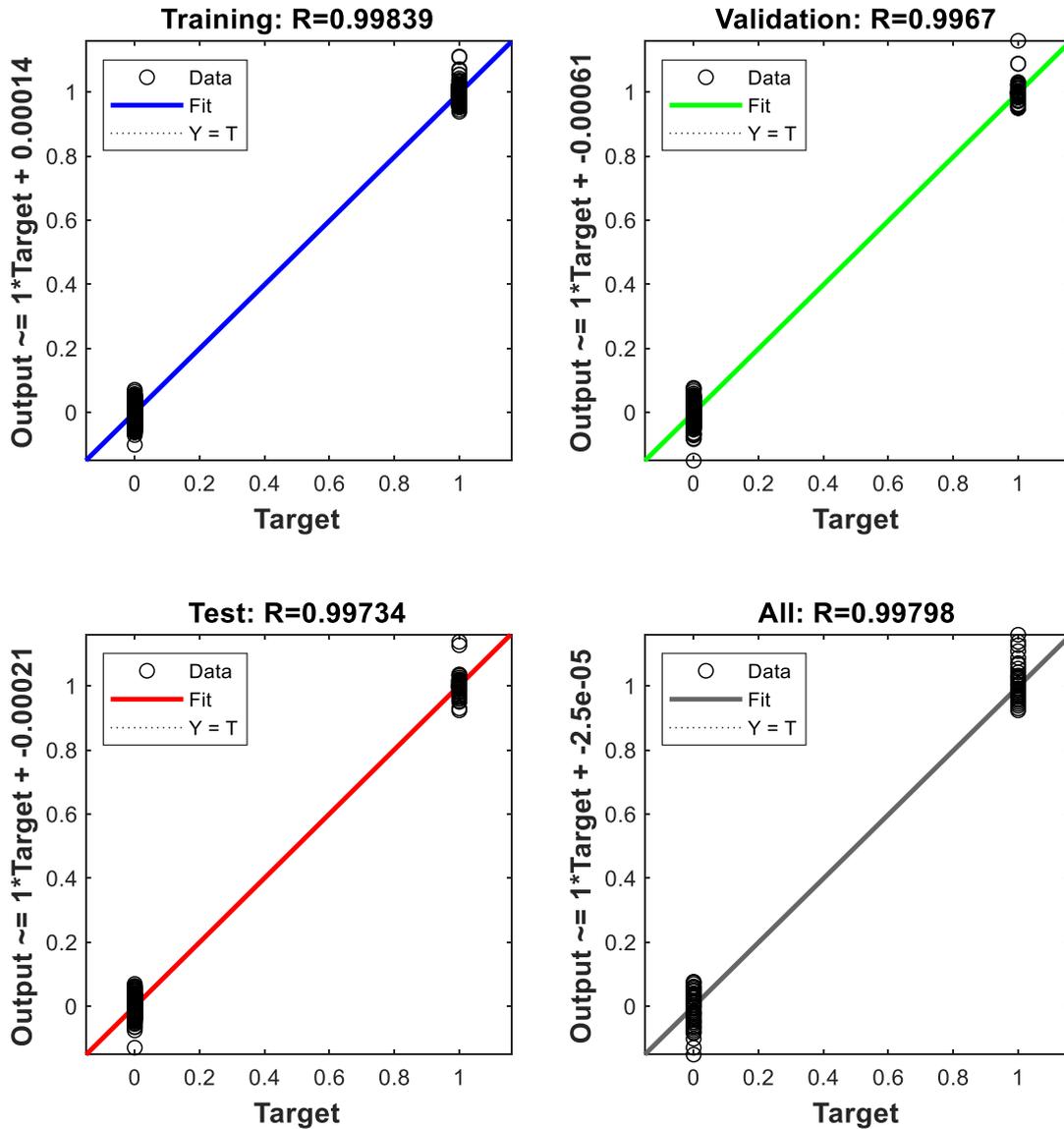


Figure 9. This is the regression results for processor graphics processing unit (GPU) with optimizer function gradient descent adaptive learning rate (GDA). There are four score values which are training score, validation score, test score, and all score. Regardless of the score types, they all appear to approach the $Y = T$ line that indicates the full percent of training correctness.

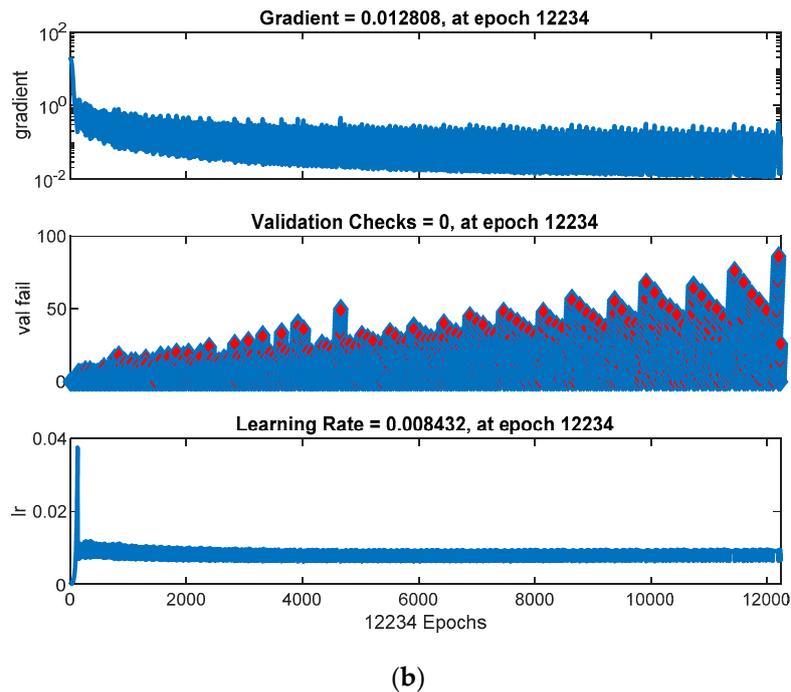
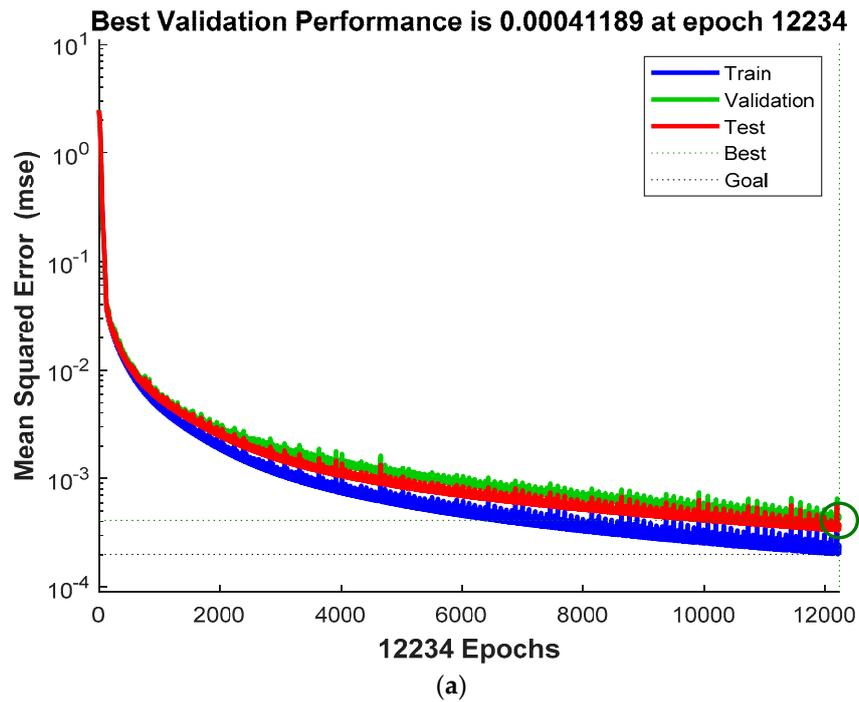


Figure 10. (a) Shows the best validation performance of processor GPU with optimizer function GDA. In addition to the convergence condition of the mean squared error, the green circle also represents whether the training could catch up with the epochs. The same concept with different settings is shown in Figures 13a, 16a and 19a; (b) This is the training state of GPU with the optimizer function GDA. Because of the adaptive learning rate, optimizer function GDA can adjust the learning rate by itself to approach the best regression but it will take a period of time. Besides, the gradient line indicates how quickly it reduces the error and converges to the goal throughout the computation and the same concept can also be seen in Figures 13b, 16b and 19b.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	0.959556	-0.00554	0.034474	-0.00227	0.010993	-0.00838	-0.06873	0.010898	0.050025	0.003914	0.038488	0.001392	-0.00816	0.00092	-0.02881
2	0.033231	0.969226	0.003449	0.027679	0.002877	0.01555	0.033494	-0.00547	-0.00388	0.012441	0.004937	0.004629	-0.01814	0.009644	0.012684
3	0.015224	-0.00381	0.950601	-0.00432	-0.01509	-0.01428	0.006626	-0.00822	0.016721	0.011652	-0.00895	-0.00784	0.021073	0.020573	-0.00242
4	-0.02112	0.013665	-0.01234	0.992827	-0.01351	-0.019	0.013935	-0.00978	-0.00807	0.007786	0.018911	-0.00589	-0.00163	0.001423	0.003416
5	-0.01745	0.001672	-0.01073	-0.01839	1.003308	0.012571	0.000763	-0.0261	0.031699	-0.00216	-0.02932	0.020486	-0.01202	-0.00137	0.017719
6	0.009759	0.00523	0.00181	0.000221	0.012172	0.99873	-0.00798	0.003529	0.003302	-0.00259	0.010552	0.007693	0.000853	-0.00499	-0.0151
7	0.018921	0.018029	-0.01766	0.042437	0.026972	-0.02176	0.950657	0.029423	-0.03141	-0.02456	0.003744	0.005421	0.008638	-0.00582	-0.01319
8	0.0053	0.024714	-0.00737	0.021217	0.023723	0.033054	-0.03925	0.976028	0.037531	0.034971	0.002423	0.014495	0.029388	-0.00668	0.008209
9	-0.03148	0.028023	0.012071	-0.00173	0.016057	0.012588	-0.0163	-0.00854	1.002924	-0.00427	0.024223	-0.00438	0.037895	0.02047	-0.00569
10	0.014183	0.003259	-0.00453	-0.00163	-0.00195	-0.00374	0.012229	0.005024	-0.00697	1.001713	0.003538	0.011327	-0.00365	0.003809	-0.00122
11	-0.00089	-0.01419	0.016591	-0.01239	-2.22E-07	-0.02491	-0.01897	-0.0349	0.018923	0.019717	0.995183	-0.00635	-0.02136	0.013378	0.039814
12	0.008644	-0.01932	0.016549	0.015516	0.010642	0.011011	0.002168	0.008801	-0.01649	0.00013	-0.01164	0.997506	-0.0035	0.015768	-0.02628
13	0.001641	0.001257	0.002662	0.003177	0.003508	-0.00238	0.015015	-0.00256	-0.01171	0.000446	-0.00726	-0.0087	0.980945	0.006655	0.003921
14	0.002633	0.010933	0.000218	-0.00264	0.00175	0.008936	-0.00517	-0.00635	0.001455	-0.00446	0.001896	-0.00358	-0.00017	0.996719	-0.00417
15	-0.00856	-0.00095	-0.00394	-0.00753	0.000121	-0.00488	0.001435	-0.00406	0.009861	0.002283	-0.00131	-1.70E-05	0.002896	0.001797	1.011678

Figure 11. The results of frog bioacoustic signal classification through GPU with the optimizer function GDA. The values represent R-scores of frog identifications. The prominent values in green blocks mean that each frog has been identified closely with the actual species, as well as in Figures 14, 17 and 20.

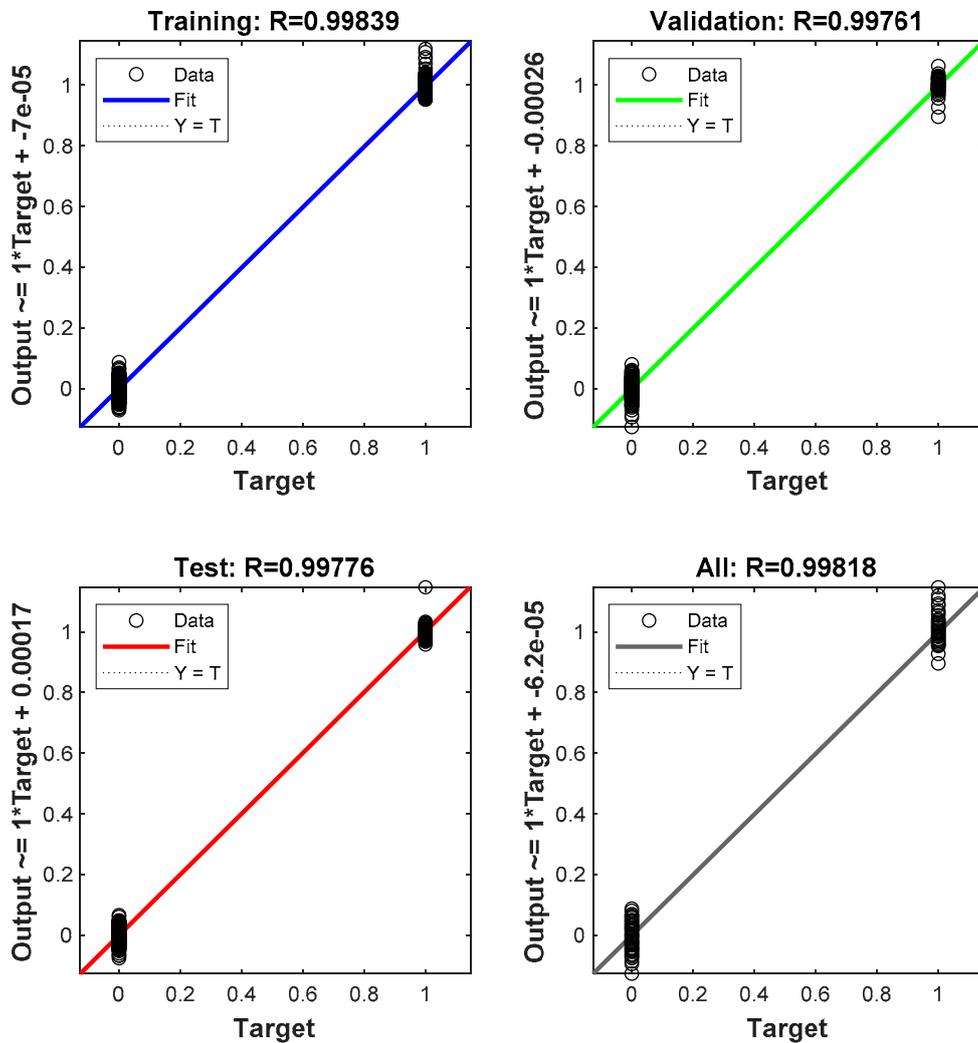
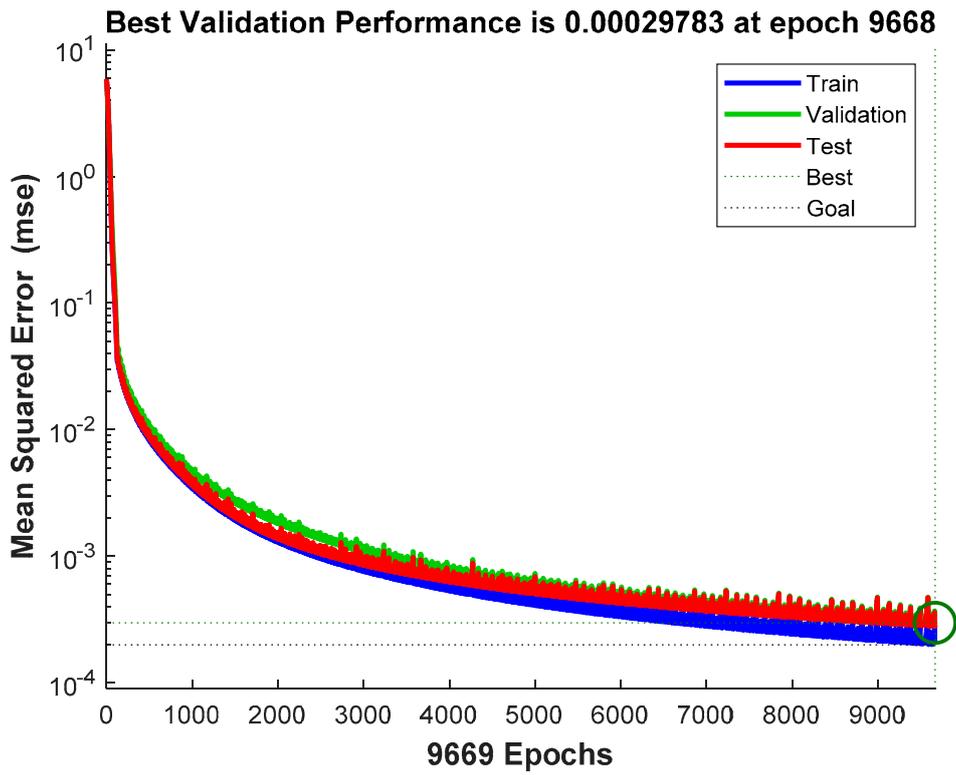
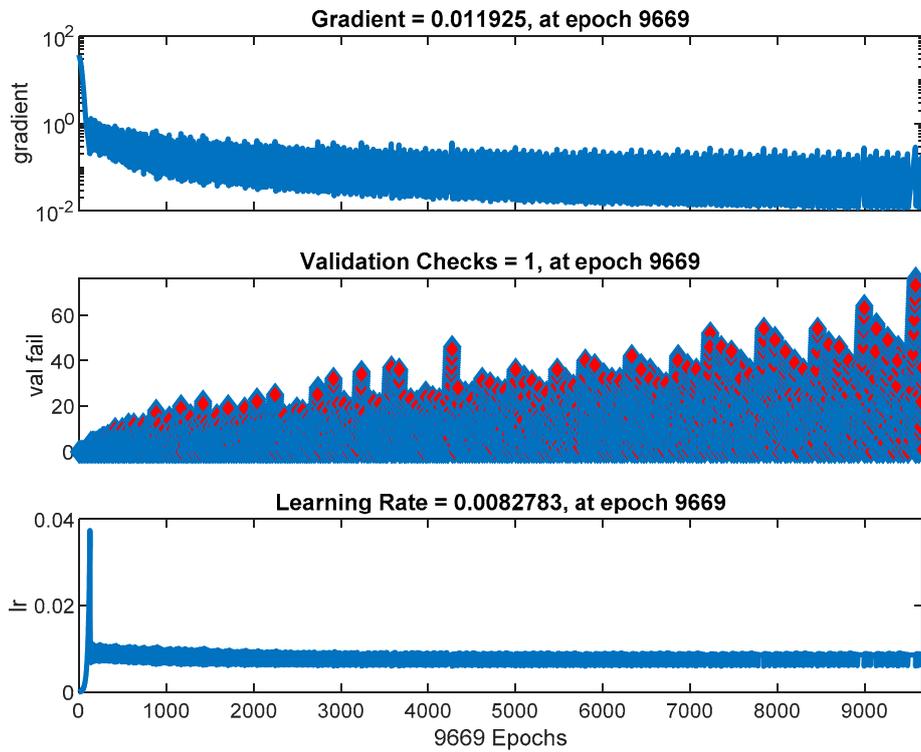


Figure 12. The regression results for central processing unit (CPU) with the optimizer function GDA.



(a)



(b)

Figure 13. (a) The best validation performance of CPU with the optimizer function GDA; (b) The training state of CPU with the optimizer function GDA.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1.024047	-0.00151	0.014597	-0.00448	0.025307	-0.00148	0.006555	-0.01555	0.037332	-0.02581	0.025264	0.043541	0.025946	0.012684	-0.02343
2	0.009558	0.954695	0.005874	-0.00334	-0.00888	-0.00829	0.036875	-0.01293	0.012189	-0.00503	-0.01254	0.003618	0.011742	-0.01968	-0.01119
3	0.007895	0.0131	0.981045	0.007867	-0.0054	-0.00872	-7.31E-05	0.002394	-0.0083	-0.00243	-0.00582	-0.01253	0.007628	-0.00616	0.002949
4	-0.01749	0.032072	-0.00737	1.005713	0.000959	-0.01998	0.026922	0.004375	0.005493	0.044549	0.000172	-0.00122	0.004923	-0.04169	0.008021
5	-0.02045	0.00499	-0.01511	-0.02324	0.985356	0.024469	-0.01455	-0.01542	0.017275	-0.02198	0.005554	-0.03083	-0.00236	-0.00979	0.001279
6	-0.00502	-0.01162	-6.39E-03	-0.00036	-0.00854	0.997262	-0.0115	-0.00997	-0.0134	0.013431	0.005843	-0.01882	0.007414	-0.00273	0.004825
7	-0.02129	0.017848	-0.02277	-0.00545	6.36E-03	-0.00373	0.986162	0.025078	0.016591	-0.00799	0.004846	-0.00083	-0.00795	-0.00621	0.007453
8	-0.02358	-0.02478	0.010178	-0.02915	0.025604	0.005669	-0.02366	0.988239	0.039406	-0.04008	0.034868	-0.01995	0.087392	-0.03556	0.011026
9	-0.01945	-0.00933	-0.01606	-0.01727	-0.01	0.007466	-0.01768	-0.00585	1.032264	0.001644	-0.00991	0.022697	0.024416	0.002104	0.004871
10	-0.00986	-0.00471	-0.0073	-0.0107	-0.00381	-0.00636	0.009855	-0.00857	0.012166	1.005361	-0.0122	0.00955	0.019453	-0.00477	0.003115
11	0.003747	-0.00547	-0.00229	-0.01658	-0.00672	-0.01082	0.013795	-0.00345	0.002424	-0.00948	1.012938	-0.01684	-0.0077	0.009446	0.016832
12	-0.0036	0.002463	-0.00455	-0.01592	0.014837	-0.00135	-0.01129	0.019942	-0.01075	-0.00229	-0.00493	1.001021	0.003449	-0.00759	-0.00414
13	0.004966	0.016946	-0.00602	0.003972	-0.00035	-0.00321	0.006631	0.008253	-0.01695	0.00469	-0.00071	0.009766	0.979656	0.004293	0.012307
14	0.001068	0.004053	0.002671	0.014258	0.001351	0.005125	-0.00263	0.021763	-0.01333	-0.01243	-0.00408	-0.00992	-0.00116	0.999029	-0.00995
15	0.004291	0.001745	-0.00368	0.002012	-0.01129	0.006304	0.017029	0.00198	-0.00473	0.002066	-0.014	0.005028	0.008312	-0.0146	0.994661

Figure 14. The results of frog bioacoustic signal classification through CPU with the optimizer function GDA. The values represent R-scores of frog identifications.

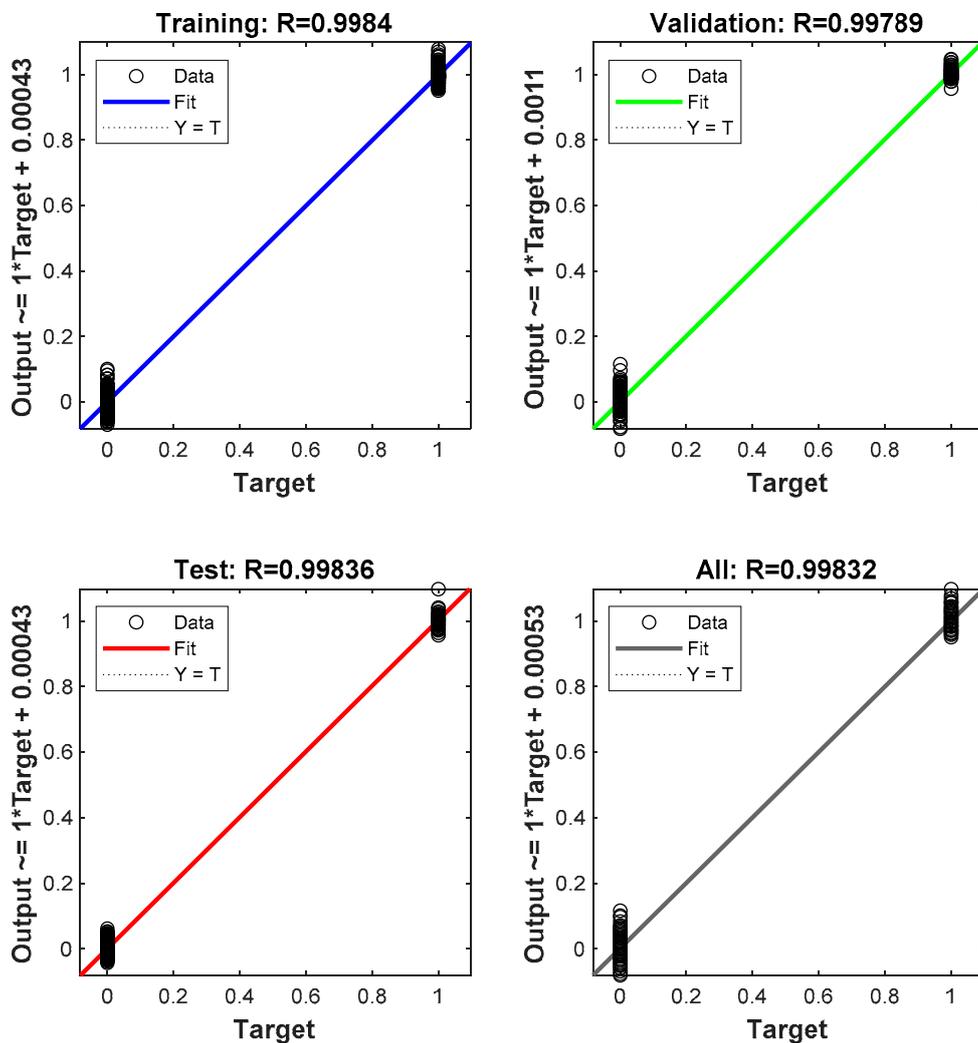


Figure 15. The regression results for GPU with the optimizer function scaled conjugate gradient (SCG).

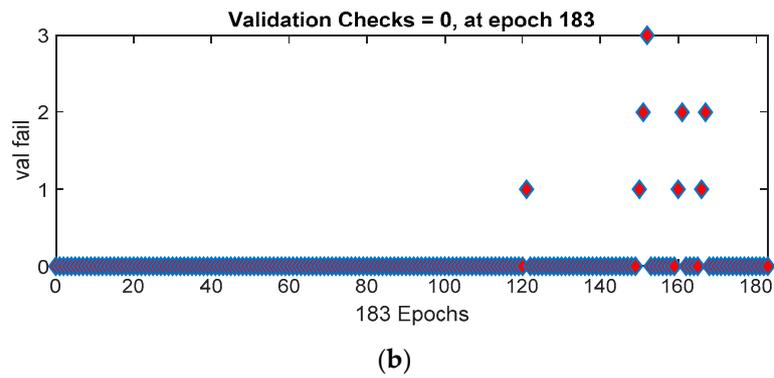
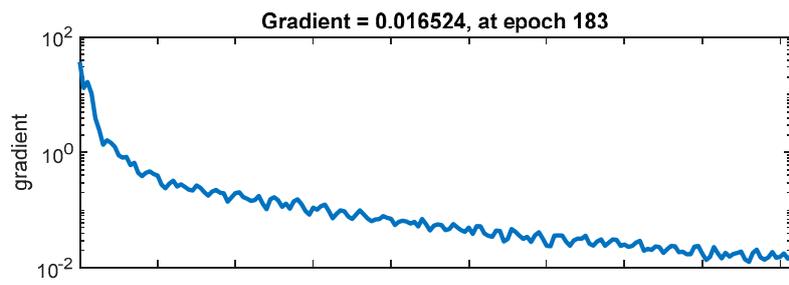
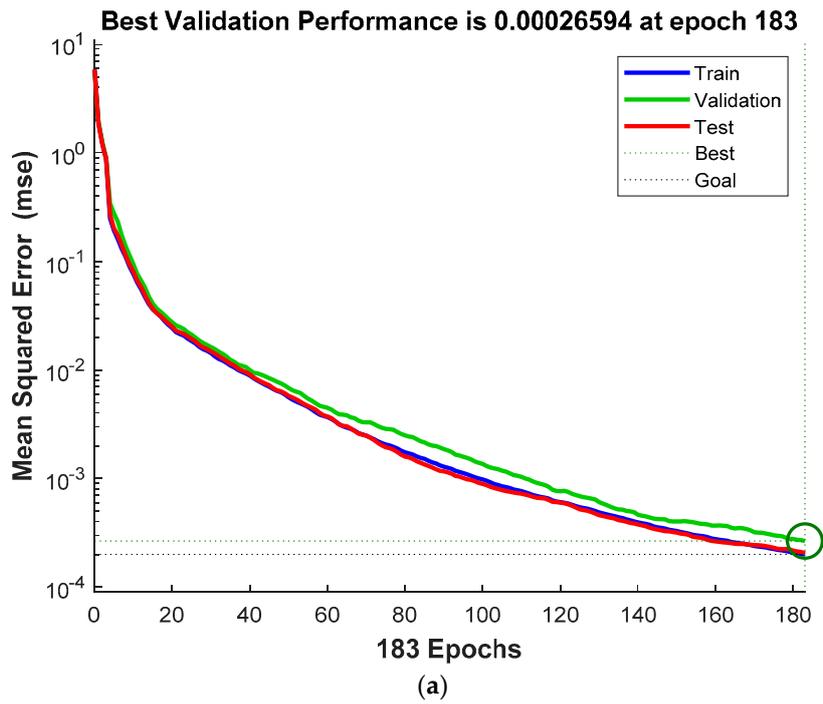


Figure 16. (a) The best validation performance of GPU with the optimizer function SCG; (b) The training state of GPU with the optimizer function SCG.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1.006721	-0.02001	0.024305	0.016201	0.051206	0.013031	0.002684	-0.01393	0.055468	-0.00894	0.013559	0.038607	0.024221	-0.00417	-0.01235
2	0.009809	0.96205	0.008096	-0.005	0.008912	-0.00696	0.039303	-0.01431	0.011916	-0.00716	-0.01547	0.001005	0.031228	-0.00974	-0.01383
3	0.002955	-0.00407	1.003749	-0.00254	0.005606	-0.00966	0.013899	0.005536	0.000708	-0.00372	0.003956	-0.01283	-0.00211	-0.01027	-0.01112
4	0.002702	0.00903	-0.00226	1.001989	0.000214	-0.01201	0.018527	0.002473	-0.00527	0.04445	-0.00598	-0.0015	0.004853	-0.02454	-0.00293
5	-0.00511	-0.0012	0.000698	-0.0199	0.997223	0.02822	-0.01164	-0.01444	-0.00215	-0.02675	-0.01087	-0.02925	-0.02651	-0.01748	-0.01369
6	-0.01027	-0.00406	-0.00475	-0.01025	0.009388	0.997317	-6.01E-06	-0.01451	-0.01097	0.004897	0.002781	-0.0181	0.004903	-0.01171	0.006413
7	-0.00373	0.018876	-0.01948	-0.01142	-0.00614	0.004548	0.992922	0.034336	-0.00257	0.006819	-0.01065	0.010722	-0.00697	-0.00603	0.007797
8	-0.01622	-0.02305	0.034781	-0.00762	0.082929	0.003913	-0.05848	1.02407	0.050655	-0.05162	0.013237	0.015286	0.101018	-0.00525	0.036785
9	-0.00292	-0.00773	-0.01349	-0.03518	0.016668	-0.00396	-0.01251	0.00256	1.045179	0.002231	-0.0084	0.029645	0.016408	-0.00062	0.023661
10	-0.01424	0.007297	-0.00687	-0.00211	-0.00055	0.008548	0.010746	-0.01817	0.003246	0.99816	-0.00143	0.005117	0.004307	-0.00449	0.01393
11	-0.00239	0.007263	-0.00051	0.008997	-0.00982	0.01071	0.003958	-0.02706	-0.0029	-0.00239	1.020559	0.008942	-0.00904	0.008736	0.005008
12	-0.00274	-0.00769	0.009869	-0.00504	0.017479	0.004689	-0.00108	-0.00093	-0.00743	-0.00481	0.002599	1.001262	-0.00204	-0.00953	0.001273
13	0.006787	0.015852	0.000683	0.005717	-0.00233	-0.00839	0.000583	0.004358	-0.01216	0.002617	0.006657	-0.00641	0.980669	0.002525	0.006342
14	0.002111	0.007579	0.001051	0.013901	0.005943	0.010477	0.001952	0.025501	-0.00424	-0.01412	0.000899	-0.00331	-0.00459	1.013055	-0.00464
15	0.00115	0.000694	-0.01452	-6.80E-05	-0.0065	0.009285	0.006705	-0.00743	-0.00442	-0.00093	-0.00549	-0.00842	0.003596	-0.01405	1.000601

Figure 17. The results of frog bioacoustic signal classification through GPU with the optimizer function SCG. The values represent R-scores of frog identifications.

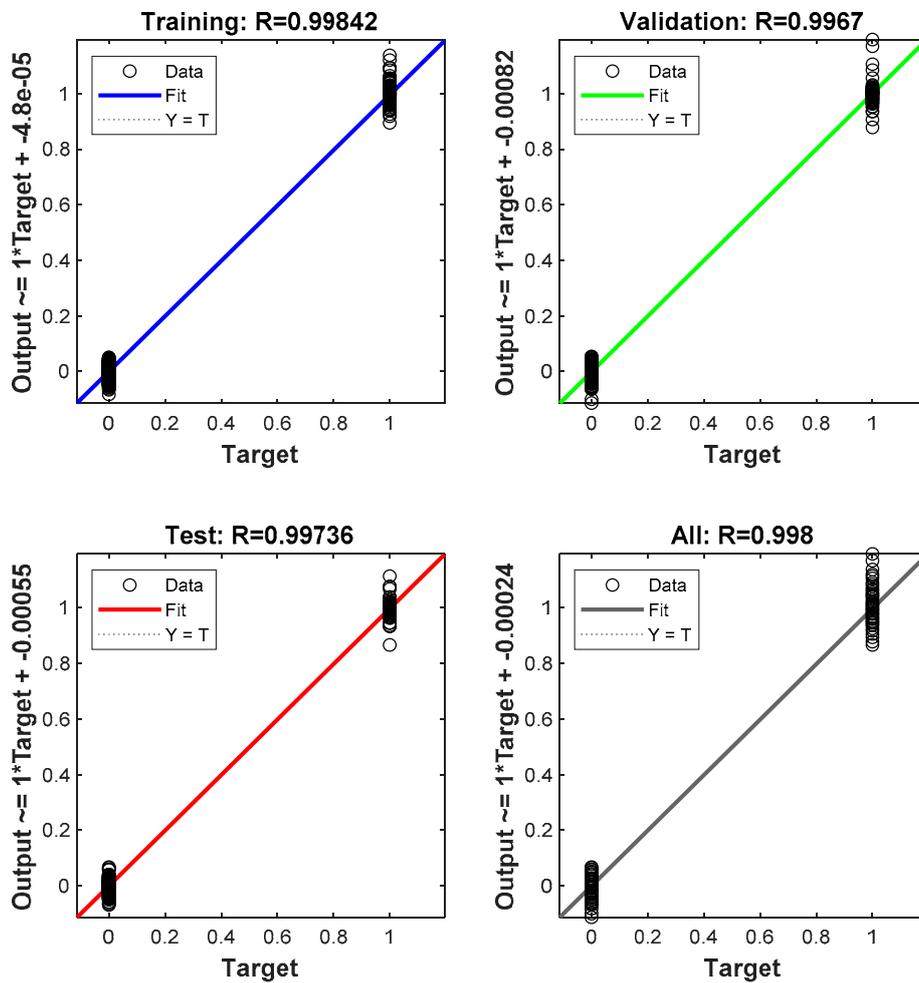
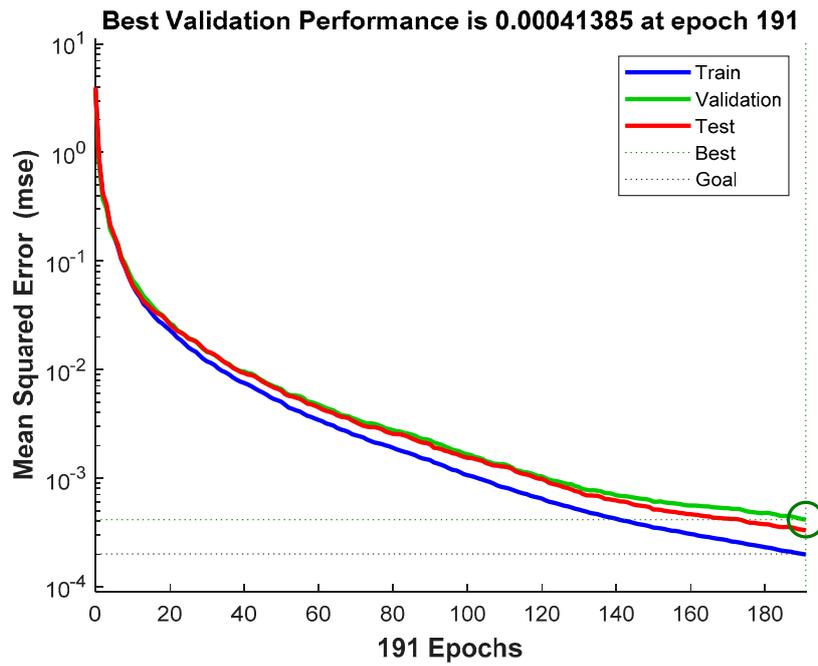
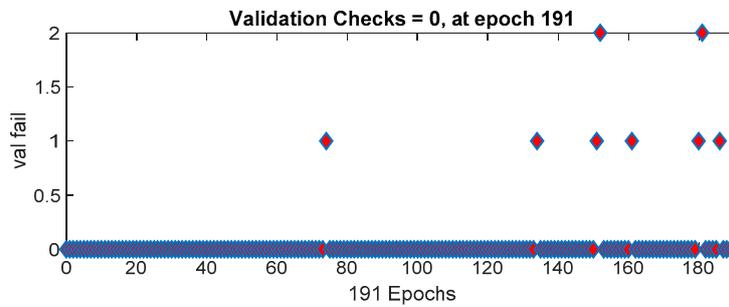
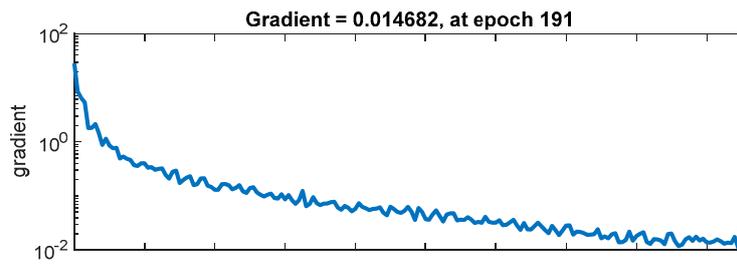


Figure 18. The regression results for CPU with the optimizer function SCG.



(a)



(b)

Figure 19. (a) The best validation performance of CPU with the optimizer function SCG; (b) The training state of CPU with the optimizer function SCG.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	0.990032	0.02714	-0.02027	0.004655	0.006414	1.65E-02	-0.05009	0.024606	0.04329	-0.00849	0.01589	0.002697	0.013134	0.022051	-0.03906
2	0.029116	0.958202	-0.01322	-0.01159	0.004224	-0.00672	0.029144	-0.00866	-0.00299	-0.00545	0.001658	-0.00906	0.000623	0.001708	0.006004
3	0.01437	-0.01851	0.935684	-0.00478	0.013608	0.01534	-0.01312	-0.02501	-0.00763	0.00644	-0.01831	-0.0123	0.002341	0.013632	0.000644
4	-0.00812	0.005034	0.002141	0.991841	0.00982	0.002363	0.001675	-0.03793	0.0056	-0.0005	-0.00061	-0.01265	0.000918	-0.00892	-0.00259
5	-0.01447	-0.01618	0.00715	0.039391	1.015307	-0.00985	-0.04838	0.022243	-0.00659	0.005674	0.01858	0.000133	0.004112	-0.01306	-0.03017
6	0.003455	0.012022	7.98E-04	-0.00111	-0.00958	1.008603	-0.0007	0.019223	0.003931	0.003043	-0.00725	-0.01401	-0.00913	0.012741	-0.00171
7	-0.0279	0.037045	0.009824	0.032752	0.03285	0.008598	0.917966	0.021975	0.012911	0.005216	-0.01045	4.49E-06	-0.00889	0.006631	0.012704
8	-0.01758	-0.06342	-0.00139	0.012624	-0.03062	-0.0222	-0.03643	1.029268	0.016508	-0.03223	0.040491	0.032132	0.031896	0.014155	0.022544
9	6.06E-03	-0.02724	-0.00816	0.019238	0.025662	0.001355	-0.0186	3.88E-05	1.054371	-0.01564	-0.00274	0.007363	0.010882	0.027647	0.015805
10	-0.01211	-0.01752	-0.00137	-0.0038	-0.00764	-0.00273	0.00822	0.008551	0.006844	1.000469	0.013691	0.00046	-0.00219	-0.00287	-0.00016
11	-0.01601	-0.01326	0.010858	0.022945	-0.01439	0.006812	0.00851	0.00977	0.025286	-0.02497	1.022227	0.019896	0.007985	-0.01295	-0.01455
12	0.016363	-0.00709	-0.00904	0.009955	0.014037	0.014325	0.010862	0.008409	0.02003	0.015101	-0.00391	1.003391	-0.00417	0.001853	-1.14E-02
13	0.002746	0.01236	0.002357	0.004169	-0.00612	-0.00652	0.018776	0.00749	-0.01054	-0.00187	-0.00253	0.001826	0.968188	-0.00337	-0.00402
14	-0.02596	-0.0177	-0.01554	0.010979	0.001698	0.006987	-0.00527	0.003683	-0.00547	-0.01683	-0.01647	-0.00995	-0.00554	0.999237	0.000374
15	-0.01304	-0.01738	0.009938	0.004262	1.46E-02	-0.00571	-0.02388	-0.00267	-0.00407	-0.00136	0.001759	-0.0138	-0.00075	0.00241	1.007256

Figure 20. The results of frog bioacoustic signal classification through CPU with the optimizer function SCG. The values represent R-scores of frog identifications.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15

Figure 21. The classification result of support vector machine shows that each frog is corresponded to the actual frog species.

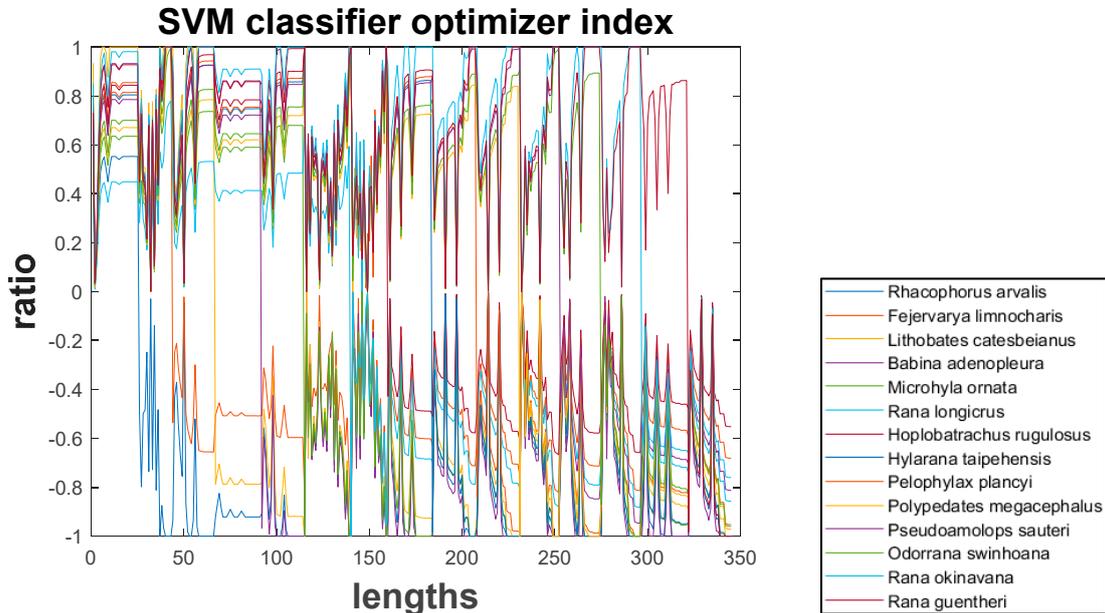


Figure 22. This is the classification diagram of support vector machine (SVM). Each frog is regarded as a class. By using the nonlinear SVM algorithm, here the first frog is chosen to be the standard point and the other frogs could be classified based on the first frog. The x-axis represents that feature values from twenty-five pre-emphasis "a" has selected with 350 points while y-axis shows the classified condition shown as a ratio. Zero ratio indicates the hyperplane. The ratio could give you the relative possibility to identify each frog (ratio in negative sign is just like positive sign).

Although the four combinations of processors and optimizers appear to have high and similar R-scores, there still exist some differences. First, due to the adaptive learning rate, optimizer function

GDA can adjust the learning rate by itself to approach the best regression. Second, in Figure 10b, it is obvious that the gradient with function GDA goes up and down, appearing to be quite unstable. However, when Figure 10b is compared with Figure 16b, the gradient with function SCG converges more stably, and Figures 10a and 16a confirm the observation. Third, in Figure 10b, the validation check raises dramatically but in Figure 16b it stays almost unchanged, meaning that the GDA function needs more iteration checks to balance the unsteady gradient so that the gradient could converge. Fourth, as shown in Table 1, the total time taken by combinations of process GPU and either of the two optimizers is smaller than that taken by other combinations, suggesting that the computation rate of processor GPU is faster than that of processor CPU. Furthermore, Table 2 compares the total time taken by neural networks with that taken by support vector machine, and the result shows that neural networks take slightly less time than support vector machine.

4. Conclusions

This study applies artificial intelligence (AI) to frog species classification of symmetry of the bioacoustics spectral. The feedforward neural networks structure is key to carry out the feature extraction and effective classification. This research uses the bioacoustic signals and the machine learning method to identify twenty-five frogs. Our study proposes an acoustic signal processing MFCC method and a classification system based on the feedforward neural networks approach.

The proposed method is to perform bioacoustic detection for categorization of animal species and use neural networks equipped with machine learning methods on the basis of training datasets to compute and develop the best model for making meaningful predictions. It is a well-suited way to understand anything that can produce sound. Through bioacoustic detection, making predictions of things that are of primary concern based on voice signals becomes feasible and meaningful. In the long term, once the method is fully developed, it is possible to improve these skills in any field.

The experimental results of this study show that two optimizer functions and two processors could effectively perform predictions and reliable analyses on different frogs. The optimizer function SCG can achieve better identification and classification. In addition, the result of this research indicates that we can directly use original data as input for machine learning. In summary, our study demonstrates that effective machine learning through big data can be reached by using the recurrent neural network (RNN) algorithm of a time-variant method. The AI algorithm helps search for distinctive and interesting features, make meaningful classification, and identify feasible feature models corresponding to various animal bioacoustics conditions.

In the future work, we will try to conduct a similar experiment by using different machine learning methods and algorithms, but the dataset will include thirty-five kinds of frogs. The wavelets algorithm is also a very useful feature extraction tool in frequency domain analysis. However, the new machine learning that the recurrent neural networks (RNN) and long short-term memory work (LSTM) methods bring can be available to apply to nature language identification. In addition to classifying the same animal, it is possible for us to advance the classification process by including many kinds of animals. As the methods of acoustic detection could be used widely, we hope that it can lead to an improvement in the medical field so that we can avoid time-consuming therapies and add some acoustic detection methods to analyze the symptoms more efficiently, such as heartbeat and the sound of flowing blood.

Author Contributions: C.-K.S. and Y.-C.C. generated the research idea, collect related information and assisted in the experiment. K.-W.C. did the experiment, integrated the materials and wrote the article. N.-Z.H. and W.-H.C. technically supported the project, supervised the experiment, polished the whole article and proofread all the details.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank the website, Frog's Word (<http://learning.froghome.org/>), for providing the sound materials for this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Gretta, T.P.; Miguel, B.A.; Johann, D.B.; Julia, B.; Timothy, C.B.; Chen, I.C.; Timothy, D.C.; Robert, K.C.; Finn, D.; Birgitta, E.; et al. Biodiversity redistribution under climate change: Impacts on ecosystems and human well-being. *Science* **2017**, *355*, 9214.
- Davis, M.B.; Shaw, R.G. Range shifts and adaptive responses to quaternary climate change. *Science* **2001**, *292*, 673–679. [[CrossRef](#)] [[PubMed](#)]
- Peter, M.N.; Sebastiaan, W.F.M. Climate change and frog calls: Long-term correlations along a tropical altitudinal gradient. *Proc. R. Soc. B* **2014**, *281*. [[CrossRef](#)]
- Xie, J.; Towsey, M.; Zhang, J.; Paul, R. Detecting frog calling activity based on acoustic event detection and multi-label learning. *Procedia Comput. Sci.* **2016**, *80*, 627–638. [[CrossRef](#)]
- Amalia, L.; Javier, R.L.; Alejandro, C.; Julio, B. Non-sequential automatic classification of anuran sounds for the estimation of climate-change indicators. *Expert Syst. Appl.* **2018**, *95*, 248–260.
- John, J.V.; Colin, T.; Michael, K.; Alex, T.; Joah, M. Applications of machine learning in animal behaviour studies. *Anim. Behav.* **2017**, *124*, 203–220.
- Kelly, R.F.; Matthew, J.S.; Mason, A.P.; Noa, P.W. The use of multilayer network analysis in animal behaviour. *Anim. Behav.* **2019**, *149*, 7–22.
- Xie, J.; Karlina, I.; Lin, S.; Michael, T.; Zhang, J.; Paul, R. Acoustic classification of frog within-species and species-specific calls. *Appl. Acoust.* **2018**, *131*, 79–86. [[CrossRef](#)]
- Ahmad, A.; Miad, F. Acoustic signal classification of breathing movements to virtually aid breath regulation. *IEEE J. Biomed. Health Inform.* **2013**, *17*, 493–500.
- Chen, Y.T. An Intelligent Nocturnal Animal Sound Identification System. Master's Thesis, National Dong Hwa University, Taiwan, July 2011.
- Amalia, L.; Jesús, G.B.; Alejandro, C.; Julio, B. Exploiting the symmetry of integral transforms for featuring anuran calls. *Symmetry* **2019**, *11*, 405.
- Wu, J.D.; Mingsian, R.B.; Su, F.C.; Huang, C.W. An expert system for the diagnosis of faults in rotating machinery using adaptive order-tracking algorithm. *Expert Syst. Appl.* **2009**, *36*, 5424–5431. [[CrossRef](#)]
- Tuomas, V.; Mark, D.P.; Daniel, P.W.E. *Computational Analysis of Sound Scenes and Events*; Springer International Publishing: Cham, Switzerland, 2017; pp. 303–333.
- Huang, C.J.; Yang, Y.J.; Yang, D.X.; Chen, Y.J. Frog classification using machine learning techniques. *Expert Syst. Appl.* **2009**, *36*, 3737–3743. [[CrossRef](#)]
- Jeffrey, A.N.; Sue, E.M.; Phyllis, J.S. A sound budget for the southeastern Bering Sea: Measuring wind, rainfall, shipping, and other sources of underwater sound. *J. Acoust. Soc. Am.* **2010**, *128*, 58–65.
- Lei, B.; Zhang, Y.; Yang, Y. Detection of sound field aberrations caused by forward scattering from underwater intruders using unsupervised machine learning. *IEEE Access* **2017**, *7*, 17608–17616. [[CrossRef](#)]
- Anshul, T.; Daksh, T.; Padmanabhan, R.; Aditya, N. Deep metric learning for bioacoustic classification: Overcoming training data scarcity using dynamic triplet loss. *J. Acoust. Soc. Am.* **2019**, *146*, 534–547.
- Anshul, T.; Padmanabhan, R. Deep archetypal analysis based intermediate matching kernel for bioacoustic classification. *IEEE J. Sel. Top. Sign. Process.* **2019**, *13*, 298–309.
- Juan, J.N.A.; Carlos, M.T.; David, S.R.; Malay, K.D.; Garima, V. Automatic classification of frogs calls based on fusion of features and SVM. In Proceedings of the Eighth International Conference on Contemporary Computing (IC3), Noida, India, 20–22 August 2015.
- Lincon, S.S.; Bernardo, B.G.; Kazuhiro, F. Classification of bioacoustic signals with tangent singular spectrum analysis. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019.
- Stavros, N. Automatic acoustic classification of insect species based on directed acyclic graphs. *J. Acoust. Soc. Am.* **2019**, *145*, 541–546.
- Kirsten, M.P.; Meah, V.L.; Joanne, M.A.N. Frogs call at a higher pitch in traffic noise. *Ecol. Soc.* **2009**, *14*, 1–24.
- Oscar, E.O.; Luis, J.V.R.; Carlos, J.C.B. Variable response of anuran calling activity to daily precipitation and temperature: Implications for climate change. *Ecosphere* **2013**, *4*, 1–12.
- Paul, S.C.; Jeff, H. Designing better frog call recognition models. *Ecol. Evol.* **2017**, *7*, 3087–3099.
- Qian, K.; Zhang, Z.; Alice, B.; Björn, S. Active learning for bird sound classification via a kernel-based extreme learning machine. *J. Acoust. Soc. Am.* **2017**, *142*, 1796–1804. [[CrossRef](#)]

26. Wang, Y.; Wei, Z.; Yang, J. Feature trend extraction and adaptive density peaks search for intelligent fault diagnosis of machines. *IEEE Trans. Ind. Inform.* **2019**, *15*, 105–115. [[CrossRef](#)]
27. Frogs' World. Available online: <http://learning.froghome.org/> (accessed on 1 January 2010).
28. Xie, J.; Michael, T.; Zhang, J.; Paul, R. Frog call classification: A survey. *Artif. Intell. Rev.* **2018**, *49*, 375–391. [[CrossRef](#)]
29. Xie, J.; Michael, T.; Zhang, J.; Paul, R. Investigation of acoustic and visual features for frog call classification. *J. Sign. Process. Syst.* **2019**. [[CrossRef](#)]
30. Marc, G.; Damian, M. Environmental sound monitoring using machine learning on mobile devices. *Appl. Acoust.* **2020**, *159*, 107041.
31. Jesús, B.A.; Josué, C.; Rohit, S.; Carlos, M.T.; Federico, B.; Adrián, G.; Alexander, V.; Mark, W. Automatic anuran identification using noise removal and audio activity detection. *Expert Syst. Appl.* **2017**, *72*, 83–92.
32. Juan, G.C.; Marco, C.; Mario, S.J.; Eduardo, F.N. An incremental technique for real-time bioacoustic signal segmentation. *Expert Syst. Appl.* **2015**, *42*, 7367–7374.
33. Sebastian, R.; Vahid, M. *Python Machine Learning*, 2nd ed.; Packt Publishing: Birmingham, UK, 2017.
34. Sumeet, D.; Xian, D. *Data Mining and Machine Learning in Cybersecurity*, 1st ed.; Auerbach Publications: New York, NY, USA, 2011; pp. 7–26.
35. Francesco, C.; Alessandro, V. *Machine Learning for Audio, Image and Video Analysis*, 2nd ed.; Springer: London, UK, 2015; pp. 99–105.
36. Lindsalwa, M.; Mumtaj, B. Voice recognition algorithms using Mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *J. Comput.* **2010**, *2*, 138–143.
37. Gong, C.A.; Su, C.S.; Chuang, Y.C.; Tseng, K.H.; Li, T.H.; Chang, C.H.; Huang, L.H. Feature extraction of rotating apparatus using acoustic sensing technology. In Proceedings of the 11th International Conference on Ubiquitous and Future Networks, Split, Croatia, 2–5 July 2019.
38. Gopi, E.S. *Digital Speech Processing Using Matlab*; Springer: Trichy, India, 2013; pp. 105–125.
39. Hiroshi, I. *Social Big Data Mining*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2015; pp. 125–135.
40. Tobias, G.; Mahmoud, K.; Robert, P.C.; Brian, C.J.M. Using recurrent neural networks to improve the perception of speech in non-stationary noise by people with cochlear implants. *J. Acoust. Soc. Am.* **2019**, *146*, 705–718.
41. Sovan, L.; Jean, F.G. *Artificial Neuronal Networks: Application to Ecology and Evolution*; Springer: Berlin, Germany, 2012; pp. 3–11.
42. James, A.A.; Edward, R. *Neurocomputing: Foundations of Research*; A Bradford Book; The MIT Press: Cambridge, MA, USA, 1989.
43. Fu, L. *Neural Networks in Computer Intelligence*; McGraw-Hill Inc.: New York, NY, USA, 1994.
44. Bhavani, T.; Latifur, K.; Mamoun, A.; Wang, L. *Design and Implementation of Data Mining Tools*, 1st ed.; Auerbach Publications: Boca Raton, FL, USA, 2019.
45. Pan, Y.P.; Liu, Y.Q.; Xu, B.; Yu, H.Y. Hybrid feedback feedforward: An efficient design of adaptive neural network control. *Neural Netw.* **2016**, *76*, 122–134. [[CrossRef](#)]
46. Dreyfus, G. *Neural Networks Methodology and Applications*; Springer: Berlin/Heidelberg, Germany, 2002.
47. Corinna, C.; Vladimir, V. Support-vector networks. *Mach. Learn.* **2007**, *20*, 273–297.
48. Guandong, X.; Zong, Y.; Yang, Z. *Applied Data Mining*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2013; pp. 112–115.
49. Almo, F. *Soundscape Ecology: Principles, Patterns, Methods and Applications*; Springer: Dordrecht, The Netherlands; New York, NY, USA, 2014.
50. Gingras, B.; Boeckle, M.; Herbst, C.T.; Fitch, W.T. Call acoustics reflect body size across four clades of anurans. *J. Zool.* **2012**, *289*, 143–150. [[CrossRef](#)]
51. Gnitecki, J.; Moussavi, Z.M. Separating Heart Sounds from Lung Sounds Accurate Diagnosis of Respiratory Disease Depends on Understanding Noises. *IEEE Eng. Med. Biol. Mag.* **2007**, *6*, 20–29. [[CrossRef](#)] [[PubMed](#)]
52. Daryush, D.M.; Matias, Z.; Feng, S.W.; Harold, A.C.; Robert, E.H. Mobile Voice Health Monitoring Using a Wearable Accelerometer Sensor and a Smartphone Platform. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 3090–3096.
53. James, M.G.; Jose, A.G.; Lam, A.C.; Stephen, R.E.; Phil, G.; Roger, K.M.; Edward, H. Restoring speech following total removal of the larynx by a learned transformation from sensor data to acoustics. *J. Acoust. Soc. Am.* **2017**, *141*, 307–313.

54. Jacob, B.; Man, M.S.; Huang, Y.A. *Springer Handbook of Speech Processing*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 161–180.
55. Sandro, S. *Introduction to Deep Learning from Logical Calculus to Artificial Intelligence*; Springer International Publishing: Berlin, Germany, 2018; pp. 51–119.
56. Marco, A.A.F. *Artificial Intelligence Emerging Trends and Applications*; Intech: London, UK, 2018; pp. 275–291.
57. Leanne, L. *Artificial Intelligence for Fashion: How AI Is Revolutionizing the Fashion Industry*, 1st ed.; Apress: New York, NY, USA, 2018; pp. 53–66, 141–144.
58. Mohamed, A.F.; Makhoul, D.; Mithun, M.; Abdelouahid, D.; Leandros, M.; Helge, J. Blockchain technologies for the internet of things: Research issues and challenges. *IEEE Internet Things J.* **2019**, *6*, 2188–2204.
59. Christopher, J.C.B. A Tutorial on Support Vector Machine for Pattern Recognition. *Data Min. Knowl. Discov.* **1998**, *2*, 121–167.
60. Pandian, V. *Artificial Intelligence Techniques and Algorithms*; Baker & Taylor: Charlotte, NC, USA, 2014; pp. 191–192.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).