Machine Learning with Applications 7 (2022) 100202

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Frog calling activity detection using lightweight CNN with multi-view spectrogram: A case study on Kroombit tinker frog

Jie Xie [a,d,e], Mingying Zhu [b,c], Kai Hu [a,d], Jinglan Zhang [f], Harry Hines [g], Ya Guo [a,d,*]

[a] Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, PR China
[b] School of Economics, Nanjing University, 22 Hankou Road, Nanjing, Jiangsu, 210093, PR China
[c] Johns Hopkins University - Hopkins-Nanjing Center for Chinese and American Studies, 162 Shanghai Road Nanjing University, Nanjing, Jiangsu, 210093, PR China
[d] School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, PR China
[e] Jiangsu Key Laboratory of Advanced Food Manufacturing Equipment & Technology, Jiangnan University, Wuxi 214122, PR China
[f] Science of Engineering Faculty, Queensland University of Technology Brisbane, Australia
[g] Department of Environment and Science, Queensland Government, Australia

ARTICLE INFO

Keywords:
Bioacoustic signal activity detection
Multi-view spectrogram
Lightweight CNN
Loss function

ABSTRACT

Frogs play an important role in ecological systems, while frog species across the globe are threatened and declining. Therefore, it is valuable to estimate the frog population based on an intelligent computer system. Due to the success of deep learning (DL) in various pattern recognition tasks, previous studies have used DL-based methods for frog call analysis. However, the performance of DL-based systems is highly affected by their input (feature representation). In this study, we develop a frog calling activity detection system for continuous field recordings using a light convolutional neural network (CNN) with multi-view spectrograms. To be specific, a sliding window is first applied to continuous recordings for obtaining audio segments with a fixed duration. Then, the background noise is filtered out. Next, a multi-view spectrogram is used for characterizing those segments, which has more distinctive information than a single-view spectrogram. Finally, a lightweight CNN model is used for the detection of frog calling activity with a twin loss, where different train and test sets are used to validate the model's robustness. Our experimental results indicate that the highest macro F1-score was 99.6 ± 0.2 and 96.4 ± 2.0 using 2016 and 2017 as the train data respectively, where CNN-GAP is used as the model with multi-view spectrogram as the input.

1. Introduction

Frog is often regarded as an "indicator species" because they are the first to be affected by environmental degradation (Woodford & Meyer, 2003). However, various reasons have led to a rapid decline including habitat loss, natural resource depletion, invasive species, climate change, and so on Colonna, Carvalho, and Rosso (2020) and Colonna, Nakamura, and Rosso (2018). Monitoring frogs is becoming ever more important. Acoustic sensors provide a highly effective way for collecting frog calls (Brodie, Allen-Ankins, Towsey, Roe, & Schwarzkopf, 2020).

Compared with traditional biodiversity data-collection methods, acoustic sensors can collect audio data over larger spatio-temporal scales automatically (Pandeya, Kim, & Lee, 2018; Wimmer, Towsey, Planitz, Roe, & Williamson, 2010; Zhao et al., 2019). Therefore, several gigabytes of compressed data can be generated by an acoustic sensor per day, which makes it ever more necessary to develop automated frog calling activity detection systems (Wimmer, Towsey, Roe, & Williamson, 2013; Zhao et al., 2017). Furthermore, the development of machine learning techniques makes it possible to build an intelligent frog monitoring system (Alzubi, Nayyar, & Kumar, 2018; Jain & Nayyar, 2020).

Detecting frog calling activity is the first step for building a frog call classification system. Here, frog calling activity detection aims to discriminate recordings having frog call from recordings only having background noise. Then, different frog species are classified from those recordings with frogs. As for animal calling activity detection, previous work has proposed various frameworks for different animal species including bird (de Oliveira et al., 2015; Jahn, Ganchev, Marques, & Schuchmann, 2017), frog (Xie et al., 2020; Xie, Michael, Zhang, & Roe, 2016; Xie, Towsey, Yasumiba, Zhang, & Roe, 2015; Xie, Towsey, Zhu,

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: xiej8734@gmail.com (J. Xie), zhumy@nju.edu.cn (M. Zhu), hukai_wlw@jiangnan.edu.cn (K. Hu), jinglan.zhang@qut.edu.au (J. Zhang), Harry.Hines@des.qld.gov.au (H. Hines), guoy@jiangnan.edu.cn (Y. Guo).

https://doi.org/10.1016/j.mlwa.2021.100202
Received 30 July 2021; Received in revised form 27 October 2021; Accepted 28 October 2021
Available online 11 November 2021

Zhang, & Roe, 2017), koala (Himawan, Towsey, Law, & Roe, 2018). In those works, only calls of interested animals are regarded as signals, others are treated as background noise. Here, the animals can belong to one species, one genus, one animal, or others. It is worth noting that frog calling activity detection can be regarded as a binary classification problem including calls of interested animals and background noise. However, some previous work aims to classify frog species based on their calls, where background noise is often assumed to be completely removed in the pre-processing step (segmentation). Therefore, only calls of different frog species will be classified.

In previous work, researchers often used various unsupervised methods for detecting frog calling activity (named as syllable segmentation) (Bedoya, Isaza, Daza, & López, 2014; Colonna et al., 2018). However, most previous segmentation methods are very sensitive to background noise, which will highly affect classification performance (Xie, Hu, Zhu and Guo, 2020). To address this problem, another method is used to consider the segmentation process as a binary classification problem (frog calling activity detection), which has been investigated in one of our previous studies (Xie, Hu, Hines et al., 2020).

Like most other members of the genus *Taudactylus*, *Kroombit tinker frog* is considered critically endangered by the International Union for Conservation of Nature, whose population is still declining. The reasons for this decline might be linked to the disease Chytridiomycosis (chytrid fungus) and habitat loss. Therefore, it is important to monitor *Kroombit tinker frog* and remap its population, which can be used for launching a conservation strategy in the future.

In this study, we aim to propose a calling activity detection system for continuous recordings of *Kroombit tinker frog*. Here, the calling activity can be from a specific frog species or a frog community, which is dependent on the used data sets. Specifically, we investigate four multi-view spectrograms including Harmonic percussive source separation (HPSS)-based, delta-based, filter-based, and repeat-based spectrograms as the feature representation of frog calling activity. Then, 2D-convolution neural network (CNN) is used for the detection of frog calling activity.

The contributions of this paper can be summarized as follows: (1) A lightweight CNN is proposed to detect frog calling activity. (2) Multi-view spectrogram is used as the feature for improving the detection performance. (3) Filtering the background noise is used to improve the performance. (4) Twin loss can increase the detection performance when compared to BCE.

The rest of the paper is organized as follows: In Section 2, we describe the related work. Section 3 contains the method including preprocessing, feature representation, and the proposed CNN architecture. Section 4 reports the experimental results. Section 6 presents conclusions and directions for future work.

2. Related work

Research in frog call analysis is attracting increasing attention with several recent papers discussing different features and classification methods. Most work focuses on frog call classification, where multiple frog species are studied including (Chen, Chen, Lin, Chen, & Lin, 2012; Huang, Yang, Yang, & Chen, 2009; LeBien et al., 2020; Tomasini, Smart, Menezes, Bush, & Ribeiro, 2017). Different from frog call classification, frog calling activity aims to recognize frog species from background noise. For frog calling activity detection, Xie et al. (2015) first extracted three acoustic indices: Shannon entropy index, spectral peak track index, and harmonic index. Then, the Gaussian mixture model was employed for anuran calling activity detection. Xie et al. (2017) first used an acoustic event detection (AED) algorithm to detect those recordings having frog calls. Then, each recording was characterized by six acoustic features. Next, multi-label learning was used to classify frog species of those recordings. Finally, the results of AED and multi-label learning were accumulated to estimate frog community calling activity and species richness. Al Bashit and Valles (2018) proposed a naive approach to build a predictor model to detect the Houston Toad mating call signature in an audio file which can be paraphrased as toad voice activity detection. Here, Mel-Filterbank and Mel-Frequency Spectral Coefficient were used for feature extraction, while the Support Vector Machine and Multi-layer Perceptron neural networks were utilized as classifiers to determine the best fit. Xie, Hu, Hines et al. (2020) first segmented continuous field recordings into segments with a sliding window. Then, different CNN architectures and various two-dimensional representations of frog calls were investigated. Recently, Gan et al. (2020) proposed to use acoustic indices for recognizing frog chorus.

In addition to frog calling activity, detection of calling activity of other animal species has been investigated. de Oliveira et al. (2015) proposed a method for acoustic activity detection of birds. Here, the spectrogram was regarded as an image and morphological filtering was used for helping the detection results. Caruso et al. (2017) first detected echolocation clicks of Delphinidae. Then, Delphinidae activity over multiple temporal scales was studied. Adavanne, Drossos, Çakir, and Virtanen (2017) investigated the detection of bird calls in audio segments using stacked convolutional and recurrent neural networks. Here, two kinds of acoustic features (dominant frequency and log Mel-band energy) and their combinations were studied in the context of bird audio detection. Himawan et al. (2018) proposed to recognize koala calling activity detection using a convolutional recurrent neural network architecture.

A summary comparison of related algorithms is shown in Table 1. We can observe that various studies have investigated animal calling activity detection. However, the investigation of feature representation is limited for deep learning-based animal calling activity detection, which might reduce the detection accuracy and the generalization ability of the proposed methods. In addition, 2D-CNN combined with multi-view spectrogram has not been used for animal calling activity detection.

3. Data and methods

In this study, we investigate multi-view spectrogram and lightweight models to detect frog calling activity, and the conceptual framework of the proposed system is shown in Fig. 1.

3.1. Data description

In this experiment, we use two 24-h frog recordings, which were collected in sites along the eastern escarpment of Kroombit Tops National park in central Queensland, Australia. Both recordings were made by Song Meters. The original sampling rate of those recordings is 16 kHz. It is worth noting that one recording was collected on 2016-12-12, and another was on 2017-12-13. To reduce the bias of our proposed model, we evaluate our method using two different training strategies (Fig. 5), where the recordings collected in different years are used as train and test data. This kind of training strategy considers the effect of background noise, which is a more realistic reflection of the performance of the classification model. In this work, two-hours labeled data from each 24-h frog recording are used for the experiment. Since the duration of the sliding window is 6 s without overlap, we obtain 1200 segments for both train and test data. The performance is evaluated using a 5-fold cross-validation (CV) strategy, which is shown in Fig. 6.

3.2. Preprocessing

For signal preprocessing, a sliding window without overlap is applied to each one-hour recording, where audio segments of fixed duration are obtained. Here, the sliding window size for generating windowed signal was set to 6 s empirically. The ground truth of each audio segment is generated based on manual annotation. The annotation is

Table 1
Summary of animal calling activity methods in previous studies. Here, GMM denotes Gaussian Mixture Model, SVM and MLP are Support Vector Machines and Multilayer perceptron, CBRNN denotes convolutional bi-directional recurrent neural network, LSTM denotes long-short time memory. MFCC and LFCC are Mel-frequency and linear-frequency Cepstral coefficients.

Study	Data	Animal species	Features	Classification method	Performance
Xie et al. (2015)	Two 12-h recordings	Anuran	Shannon entropy, spectral peak track, harmonic index, and oscillation index	GMM	Precision: 63.82%, 79.01%, 75.21%, 83.01%
Xie et al. (2017)	342 10-s recordings	Frog	Linear predictive coding coefficients, MFCC, LFCC, acoustic complexity index, acoustic diversity index, acoustic evenness index	Multi-label learning	Accuracy: 72%
Al Bashit and Valles (2018)	2075 Training and 1351 test samples	Houston Toad	MFCC	SVM and MLP	Accuracy: 98.07%
Xie, Hu, Hines et al. (2020)	Four 1-h recordings	Frog	Mel-spectrogram and constant-q spectrogram	CNN	F1-score: 98.14%, 82.12%, 71.38%
Gan et al. (2020)	Two 24-h recordings	Frog	Twelve spectral indices	SVM, RF and KNN	Accuracy: 82.9%
de Oliveira et al. (2015)	93 recordings with ~45 min of sounds from target species	Bird	Spectrogram	Threshold-based	Accuracy: 56.4% and 41.3%
Caruso et al. (2017)	5500 5-min recordings	Delphinidae	Teager-Kaiser Energy Operator	Threshold-based	Accuracy: 79% (2005) 86% (2006)
Adavanne et al. (2017)	freefield1010 and warblr	Bird	Dominant frequency and log mel-band energy	CBRNN	AUC: 88.1%
Himawan et al. (2018)	3.6 h of koala calls as positive samples and 7 h of other audio clips as negative samples	Koala	Constant-q spectrogram	CNN-LSTM	AUC: 87.46%

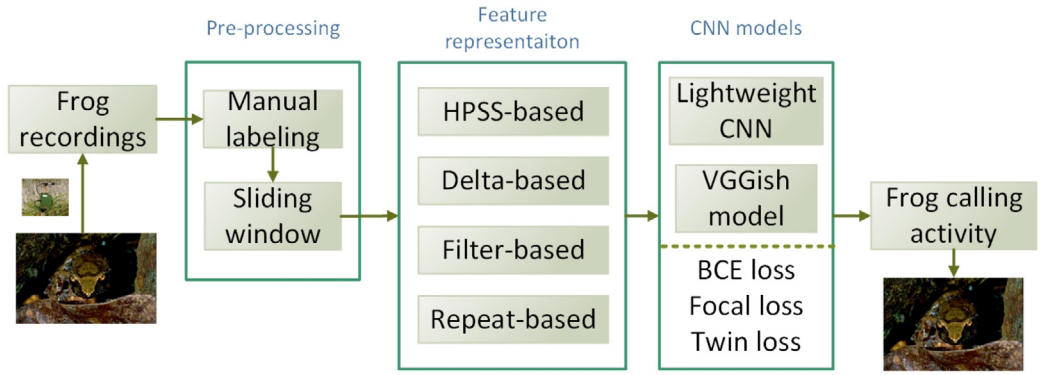


Fig. 1. Conceptual framework of our proposed frog calling activity detection system.

generated based on two trained students using both audio and visual information. Specifically, the recording is first opened by version 2.3.0 of Audacity(R) recording and editing software. Then, two students listen to the audio and watch the spectrogram to determine if the frog call exists. If the audio segment has frog calls, it will be assigned with the label 'one', otherwise label 'zero' will be given. It is worthwhile noting that when both students have different labels for an audio segment. An external frog expert will help determine the final label.

3.3. Feature representation

For the windowed signal, it is naturally used as the one dimensional (1D) representation (waveform) (Xie, Hu, Zhu et al., 2020). In addition, two-dimensional (2D) representation (spectrogram) has been widely used for frog call analysis. However, the used spectrogram is often single-view, which might lead to information loss. In this study, a multi-view spectrogram is used as the input for CNN. Specifically, various types of multi-view spectrograms are investigated.

First, the audio signal is downsampled to a quarter of the original sampling rate (4 kHz). Hence, f_{max} the highest frequency analyzed, set to half of the new sampling rate (3.5 kHz). The transform is computed with f_{min}, the lowest frequency analyzed, set to 2 kHz. The visualization of original and filtered spectrograms is shown in Fig. 2.

For a windowed signal $x(n)$, linear-scaled spectrogram is generated using short-time Fourier transform (STFT), which is defined as

$$X[f,l] = \sum_{n=-N/2}^{N/2-1} x[n+l\cdot r]\cdot w[n]\cdot e^{-j\frac{2\pi fn}{N}} \quad (1)$$

Here, $w[n]$ is the sliding window function, N is the window length, l is the frame number, r is the hop size, f and l are frequency and

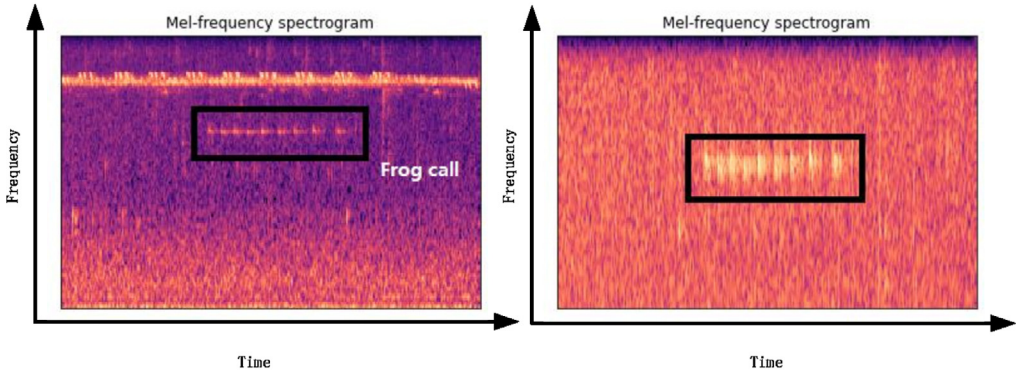


Fig. 2. Original (left) and filtered (right) spectrograms.

time index in STFT domain. Here, the linear-scaled spectrogram is the squared magnitude of the STFT, which gives the power of a sound for a specific frequency and time in the third dimension (Huzaifah, 2017).

After getting the linear-scaled spectrogram $X[f,l]$, Mel-scaled spectrogram $X[f_{mel},l]$ can be obtained by applying a set of M triangular filters in the Mel scale and then the logarithm of the resulting signal is computed. Frequencies in Hz can be converted to Mel scale as follows:

$$f_{mel} = 2595 \cdot log_{10}(1 + \frac{f}{700}) \tag{2}$$

Here, f_{mel} is the calculated Mel scale frequency, f is the standard frequency.

3.3.1. HPSS-based spectrogram

Harmonic percussive source separation (HPSS) aims to decompose an audio signal into its harmonic and percussive components (Driedger, Müller, & Disch, 2014; Fitzgerald, 2010). For HPSS, we use the power spectrogram $X[f,l]$ rather than the log-power spectrogram. Then, $H[f,l]$ and $P[f,l]$ are defined as an element of H the harmonic power spectrogram and an element of P the percussive power spectrogram. The cost function is given as follows:

$$J(H,P) = \frac{1}{2\sigma_H^2}\sum_{f,l}(H_{f,l-1} - H_{f,l})^2 + \frac{1}{2\sigma_P^2}\sum_{f,l}(P_{f,l-1} - P_{f,l})^2 \tag{3}$$

where σ_H and σ_P are parameters for controlling the weights of the harmonic and percussive smoothness respectively. The cost function is further subject to the additional constraints that

$$H_{f,l} + P_{f,l} = W_{f,l} \tag{4}$$

$$H_{f,l} \geq 0, P_{k,l} \geq 0 \tag{5}$$

Both harmonic and percussive power spectrograms are obtained using the *Librosa* library with the default parameter setting (McFee et al., 2015). After obtaining $H[f,l]$ and $P[f,l]$, the multi-view spectrogram is represented as

$$X_{multi-view-H}[f,l] = \{X[f,l], H[f,l], P[f,l]\} \tag{6}$$

3.3.2. Delta-based spectrogram

The delta-based spectrogram is obtained based on Savitzky-Golay filtering (Press & Teukolsky, 1990). The first and second-order based delta spectrograms are combined with the log-power spectrogram as follows:

$$X_{multi-view-D}[f,l] = \{X[f,l], \Delta X[f,l], \Delta\Delta X[f,l]\} \tag{7}$$

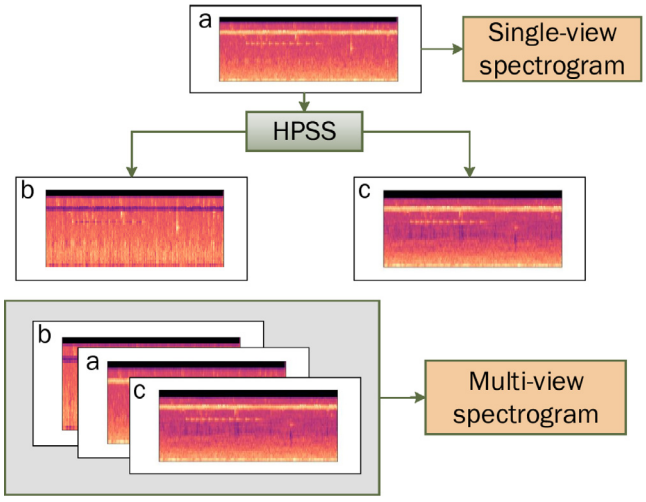


Fig. 3. An example of single-view and multi-view spectrograms. Here, HPSS can be replaced by other operations.

Table 2
A comparison of multi-view spectrograms.

Input	Technique	Spectrogram	Dimension
Power spectrogram	Median-filtering	HPSS-based	120*K*3
Log-power spectrogram	Savitzky-Golay filtering	Delta-based	
Log-power spectrogram	Filtering by nearest-neighbors	Filter-based	
Log-power spectrogram	Repeat 3 times	Repeat-based	

3.3.3. Filter-based spectrogram

The filter-based spectrogram is generated by replacing each data point (e.g, spectrogram column) using its nearest neighbors in feature space. The method is primarily developed for de-noising spectrogram. Here, we aim to use different denoised spectrograms for constructing the multi-view spectrogram.

$$X_{multi-view-F}[f,l] = \{X[f,l], X^1_{denoise}[f,l], X^2_{denoise}[f,l]\} \tag{8}$$

Here, $X^i_{denoise}$ denotes the denoised spectrogram.

In addition, we also simply repeat the spectrogram three times and construct the multi-view spectrogram as follows

$$X_{multi-view-R}[f,l] = \{X[f,l], X[f,l], X[f,l]\} \tag{9}$$

An example for single-view and multi-view spectrograms is shown in Fig. 3. For the three multi-view spectrograms, Table 2 gives the comparison in terms of input and techniques.

Table 3
Our CNN architecture. *BN*: Batch Normalization, *ReLU*: Rectified Linear Unit, *K* denotes the number of frame per recording. Here, 120 is the number of Mel-bands for generating the spectrogram. *C* denotes the channels which can be 1 or 3.

Input 120 × K × C
3 × 3 Conv(pad-1, stride-1)-32-ReLU 4 × 2 Max-Pooling + Drop-Out(0.5)
3 × 3 Conv(pad-1, stride-1)-64-ReLU 4 × 2 Max-Pooling + Drop-Out(0.5)
3 × 3 Conv(pad-1, stride-1)-128-ReLU 4 × 2 Max-Pooling + Drop-Out(0.5)
GAP
Dense(1024) + Drop-Out(0.2)
2-way Soft-Max

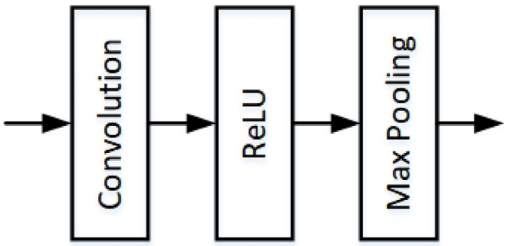


Fig. 4. Block diagram of the local feature learning block (LFLB). Here, ReLU denotes rectified linear unit. Specifically, the first represents the 2D convolution with a kernel of size 3 × 3 and stride 1 × 1; the second represents the 2D pooling with a kernel of size 4 × 2 and stride 4 × 2.

Table 4
The VGGish model. *BN*: Batch Normalization, *ReLU*: Rectified Linear Unit, *K* denotes the number of frame per recording. Here, 120 is the number of Mel-bands for generating the spectrogram. *C* denotes the channels which can be 1 or 3.

Input 120 × K × C
3 × 3 Conv(pad-1, stride-2)-32-BN-ReLU 3 × 3 Conv(pad-1, stride-2)-32-BN-ReLU 2 × 2 Max-Pooling + Drop-Out(0.2)
3 × 3 Conv(pad-1, stride-2)-64-BN-ReLU 3 × 3 Conv(pad-1, stride-2)-64-BN-ReLU 2 × 2 Max-Pooling + Drop-Out(0.2)
3 × 3 Conv(pad-1, stride-2)-128-BN-ReLU 3 × 3 Conv(pad-1, stride-2)-128-BN-ReLU 2 × 2 Max-Pooling + Drop-Out(0.2)
GAP
Dense(512) + Drop-Out(0.2)
2-way Soft-Max

3.4. Models

Our CNN architecture is made up of three local feature learning blocks, which are made up of one convolutional layer, one ReLU layer, and one max-pooling layer, as shown in Fig. 4. The architecture consists of 3 convolutional layers with a receptive field of 3 × 3, where the rectified linear unit is used as the activation function. Then, a max pooling operation is added for every convolutional layer. Dropout is employed in convolutional layers with a rate of 0.5 to address overfitting. The CNN is optimized jointly using a backpropagation algorithm.

A softmax layer with two nodes is used (one for frog call and one for background noise). The network is trained using Adam optimizer with a learning rate of $1e^{-3}$, and a batch size of 32. Table 3 shows CNN architecture in details.

In addition to the proposed lightweight CNN architecture, a VGGish model is employed for the comparison (Table 4).

3.5. Loss function

3.5.1. Cross-entropy loss

Cross-entropy (CE) loss is often used as the basic loss function for training a CNN, which can ensure the basic classification ability of the network. The CE loss is defined as:

$$L_c = \frac{1}{M}\sum_{k=1}^{K}\sum_{i=1}^{M} y_i^k log(\frac{e^{s_{y_i}}}{\sum_{j=1}^{n} e^{s_j}}) \tag{10}$$

where M is the number of samples, K denotes the number of classes, n is the dimension of the extra fully connected output layer, y_i represents the label of the ith sample, s_j represents the jth element of the classification score vector s.

However, it has been demonstrated in one previous study that CE loss can correctly classify simple samples but is powerless for relatively difficult samples (Cheng, Li, Wu, & Ngan, 2020). The reason is that CE loss does not constrain in-class compactness and inter-class separation which makes the discriminability of learned deep features less powerful. Therefore, it is worth investigating other loss functions for learning better deep features and further improving the final classification performance.

3.5.2. Focal loss

Focal loss is initially proposed to address the problem of extreme imbalanced classification (Lin, Goyal, Girshick, He, & Dollár, 2017). The traditional CE loss for binary classification is defined as follows:

$$CE(p,y) = \begin{cases} -log(p) & if\ y = 1 \\ -log(1-p) & otherwise \end{cases} \tag{11}$$

where $p \in [0,1]$ is the estimated probability for the class. For binary classification, the value of y is denoted by $0, 1$, and p ranges from 0 to 1. Then, the CE loss function can be defined as follows:

$$CE(p,y) = CE(P_i) = -log(P_i) \tag{12}$$

where

$$P_i = \begin{cases} p & if\ y = 1 \\ 1-p & otherwise \end{cases} \tag{13}$$

To deal with those samples that are difficult to be classified, a modulating factor, $(1-P_i)^\gamma$ is added to the CE loss. A small weight is assigned to easily classified samples, and CNN pays more attention to those difficult samples. The focal loss function is defined as follows:

$$L_f = -\alpha_i(1-P_i)^\gamma log(P_i) \tag{14}$$

where $(1-P_i)^\gamma$ denotes the modulation factor and $\gamma > 0$. The current focal loss function is for binary classification.

For the loss function, we use a combination of binary cross-entropy loss (BCE) and binary focal loss, which is defined as

$$Loss = \gamma L_f + (1-\gamma)L_c \tag{15}$$

Here, L_{BCE} denotes the binary cross-entropy loss, L_f denotes the binary focal loss, and γ is a hyperparameter that controls the relative weight of the terms.

4. Experiments

4.1. Experiment setup

Our code implementation is based on Python 3.8.5 and the deep learning was implemented using Keras 2.4.3 (Chollet et al., 2015) with TensorFlow 2.4.1 (Abadi et al., 2016) backend. All experiments were run on NVIDIA GeForce GTX 1080.

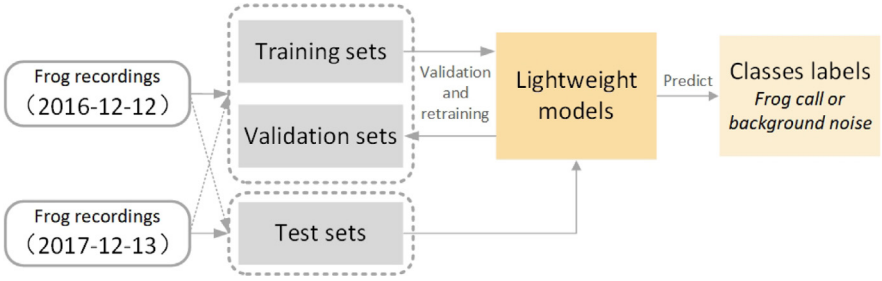


Fig. 5. Two different training strategies with different train and test data. For the train data, it is split into 5 folds. Here, four folds are used as the training set and the rest is the validation set.

Fig. 6. Schematic representation of training and validation scheme employed in the 5-fold cross-validation procedure.

Table 5
A summary of parameters for Mel-spectrogram and CNN.

Preprocessing		Model	
Slinging window	6 s	Optimizer	Adam
Mel-spectrogram		Learning rate	0.001
Frame size	40 ms	Loss function	Twin loss BCE loss Focal loss
Frame overlap	20 ms	Batch size	32
Number of band	120	Epoch	200

For the Mel-spectrogram, it is obtained using a 40 ms frame size with an overlap of 20 ms and 120 bands. For the detection, our proposed 2D-CNN model is trained using Adam optimizer with a learning rate of 10^{-3}. The loss function consists of binary cross-entropy, focal loss, and twin loss. The batch size is 32 samples and the network is trained with 200 epochs. In addition, early stopping is used to prevent over-fitting and no data augmentation is done for all networks. A summary of parameters for feature generation and model is shown in Table 5.

4.1.1. Evaluation rule

For each classifier output, a label is assigned to each sliding window. Then, the recordings collected in two different years are used as train and test sets, respectively. The performance of frog calling activity detection is evaluated by accuracy and F1-score, which are defined as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

$$F1\text{-}score = \sum_{i=1}^{n} 2 \cdot \frac{precision(i) \cdot recall(i)}{precision(i) + recall(i)} \tag{17}$$

where $F1\text{-}score$ denotes the F1-score of one class, $precision$ and $recall$ are defined as $\frac{TP}{TP+FP}$ and $\frac{TP}{TP+FN}$ respectively, TP is true positive, TN is true negative, FP is false positive, FN is false negative. i is the class index.

In addition to accuracy and F1-score, a precision–recall curve is used to evaluate classifier performance. For the PR curve, the area is often used for model comparison, where a higher area value indicated better performance.

Table 6
Detection performance using binary cross-entropy and single-view spectrogram with 5-fold test data (Mean value ± Standard deviation). Here, the train data is collected in 2016.

Fold name	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Fold-1	82.92	90.24	65.94	69.06
Fold-2	82.75	90.65	65.50	68.51
Fold-3	83.17	90.83	66.33	69.58
Fold-4	80.67	85.20	62.22	64.04
Fold-5	90.67	93.40	81.89	85.81
Mean ± Std	84.03 ± 3.43	90.06 ± 2.67	68.38 ± 6.91	71.4 ± 7.47

Table 7
Detection performance using binary cross-entropy and single-view spectrogram with 5-fold test data (Mean value ± Standard deviation). Here, the train data is collected in 2017.

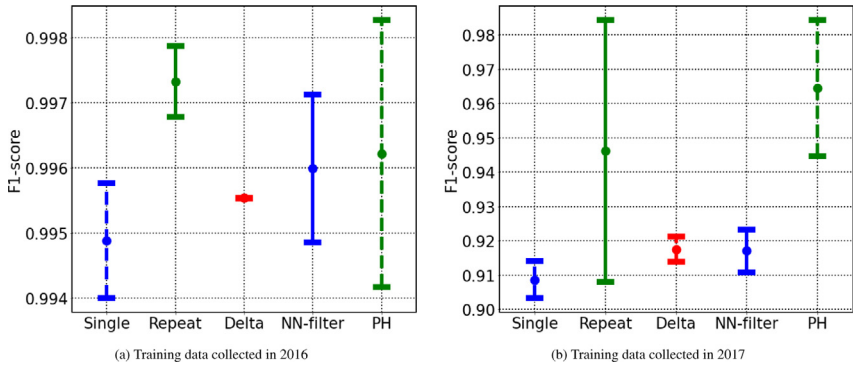
Fold name	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Fold-1	78.17	70.07	74.62	71.51
Fold-2	75.75	70.45	78.44	71.29
Fold-3	81.58	76.31	86.29	78.02
Fold-4	76.75	71.96	80.86	72.77
Fold-5	78.17	39.08	50.00	43.87
Mean ± Std	78.08 ± 1.97	65.58 ± 13.43	74.04 ± 12.6	67.49 ± 12.06

4.2. Experimental results

Tables 6 and 7 presents the performance values of the CNN model using a 5-fold CV strategy. The CNN network can reach 84.03 ± 3.43 average accuracy with a 5-fold CV. Here, the loss function and feature representations are binary cross-entropy and single-view full spectrogram. In the end, we conclude that the CNN model using binary cross-entropy and single-view full spectrogram did not reach the desired level of success. In addition, when the training data is collected in 2016, the average F1-score is 71.4 ± 7.47, which is higher than the performance using data collected in 2017 as the train data.

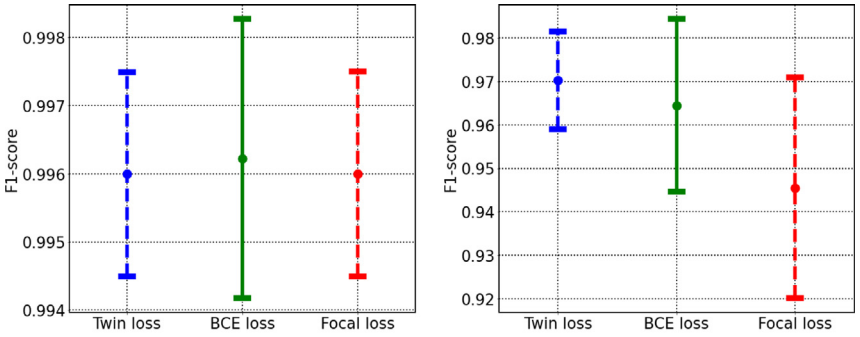
Since the peak frequency of interested frog calls is around 2.7 kHz, we further select the filtered spectrogram as the input for the classification. The use of filtered spectrogram can greatly reduce the effect of background noise. The detection F1-score using filtered and original spectrogram is shown in Table 8. From the table, we can find that the detection accuracy using filtered spectrogram (2016) can be up to 99.63 ± 0.04, which is significantly higher than the original spectrogram (84.03 ± 3.43) at 1% level (t = 32.84, p = 0.00). However, the detection accuracy using filtered spectrogram (2017) is 94.23 ± 0.24, which can be further improved but is still significantly higher than the original spectrogram at 1% level (t = 29.08, p = 0.00).

To further improve the detection performance, a single-view spectrogram is replaced by a multi-view spectrogram. The detection result is shown in Fig. 7. From the figure, we can find that the f1-score using multi-view spectrogram is higher than the single-view spectrogram. Among those multi-view spectrograms, repeat- and PH-based spectrograms can achieve a higher f1-score, but the difference is not significant.



(a) Training data collected in 2016 (b) Training data collected in 2017

Fig. 7. Detection F1-score using single-view and multi-view spectrograms.



(a) Training data collected in 2016 (b) Training data collected in 2017

Fig. 8. Detection F1-score using various loss functions.

Table 8
Detection performance using binary cross-entropy and single-view spectrogram with 5-fold test data (Mean value ± Standard deviation).

	Spectrogram	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
2016	Original	84.03 ± 3.43	90.06 ± 2.67	68.38 ± 6.91	71.4 ± 7.47
	Filtered	99.63 ± 0.04	99.6 ± 0.08	99.42 ± 0.03	99.51 ± 0.05
2017	Original	78.08 ± 1.97	65.58 ± 13.43	74.04 ± 12.6	67.49 ± 12.06
	Filtered	94.23 ± 0.24	94.0 ± 0.22	88.75 ± 0.89	91.03 ± 0.49

Table 9
Detection performance using different CNN models.

Model	2016	2017	Parameter
CNN-GAP	99.6 ± 0.2	96.4 ± 2.0	93 k
CNN-FC	99.6 ± 0.3	93.0 ± 2.2	102 k
VGGish	99.5 ± 0.3	93.4 ± 6.3	355 k

Finally, we explore different loss functions for better performance. Fig. 8 shows the comparison among BCE loss, focal loss, and twin loss (a combination of BCE loss and focal loss), where PH-based spectrogram is selected as the input. It is observed that twin loss can achieve the highest f1-score when the training data is collected in 2017, For the twin loss, the parameter γ is set at 0.6. Fig. 9 shows the confusion matrix of the best performing model. The training and validation accuracy and loss graphs of the best performing model for a fold are given in Fig. 10, where only 23 epochs are selected with early stopping. From the figure, we can conclude that the model we used is lightweight and the feature representation (multi-view spectrogram) can well discriminate frog calls from the background noise. Therefore, we learn most of the important stuff during the first few epoch.

5. Discussion

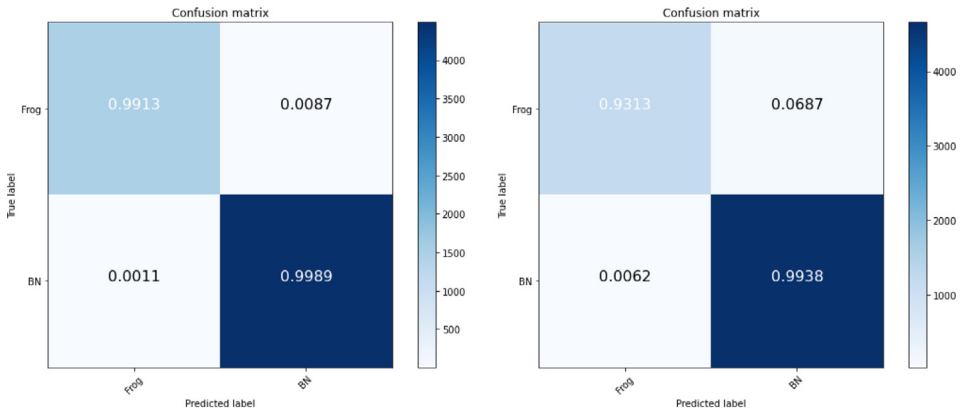
From the experiment, we can find that the performance using filtered spectrogram is better than full spectrogram. The use of filtered spectrogram can significantly reduce background noise (e.g., cricket calls, bird calls, vehicles). The reason can be summarized as follows: (1) The spectrum information of targeted frog call ranges from 2.5 kHz to 3 kHz; (2) we only focus on the detection of one frog species, rather than multiple frog species, there are no overlapped calls in the frequency domain.

A multi-view spectrogram can increase the overall performance when compared to a single-view spectrogram. Compared to the single-view spectrogram, multi-view spectrogram has more distinctive information for discriminating frog calling activity from the background noise. Finally, the use of twin loss can increase the detection performance when compared to BCE.

Few studies have investigated frog calling activity detection. In this study, we compare the proposed method with other CNN models that have been used in previous studies. The first model is the same but GAP is replaced by the fully connected layer (Himawan et al., 2018). The second model is the VGG style model, which has been used in Xie, Hu, Zhu, Yu, and Zhu (2019). The detection results are shown in Table 9. From the Table, we can observe our proposed method achieves the best performance with the fewest parameters.

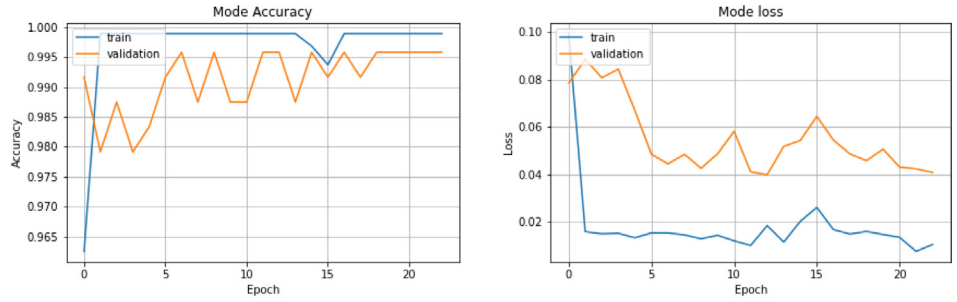
To further investigate the detection of frog calling activity, Fig. 11 shows the PR curve using the best performing model. We can find that the area of background noise is larger than the frog calling activity for both training strategies. Comparing the two plots, using the data collected in 2016 for the model can achieve better performance.

Since the proposed detection system is developed for a specific frog species, we can observe that the spectrogram pattern of *Kroombit tinker frog* ranges from 2.5 kHz to 3 kHz which makes filtering effective for improving the final detection performance. However, it might be difficult to obtain a performance improvement using filtering for multiple frog species classification due to call overlap.



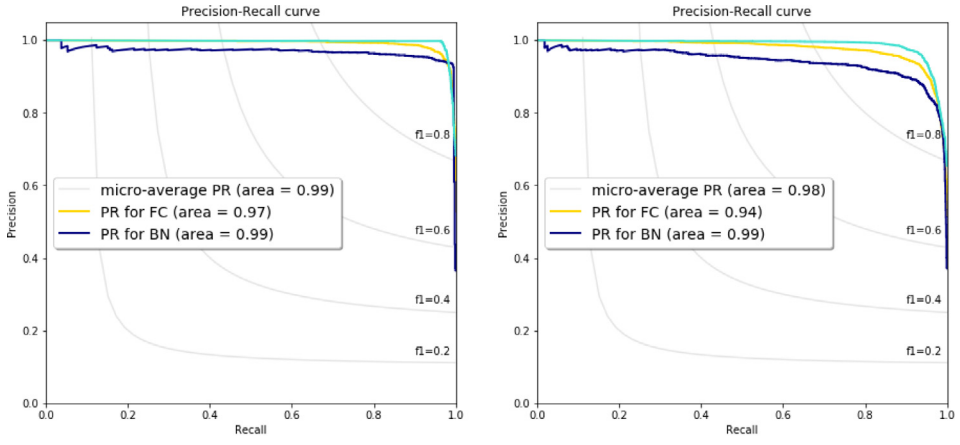
(a) Training data collected in 2016 (b) Training data collected in 2017

Fig. 9. Confusion matrix of the best performing model.



(a) Accuracy (b) Loss

Fig. 10. The training accuracy and loss of the best performing model obtained during 200 epochs with early stopping: (a) Accuracy, and (b) Loss.



(a) Train data collected in 2016 (b) Train data collected in 2017

Fig. 11. PR curves of the best performing model, whereas the micro-average PR aggregates the contributions of all classes to compute the average. Here, FC and BN denote frog calling activity and background noise, respectively.

Comparing Tables 6–8, we can find that multi-view spectrogram can improve the performance. From Table 9, we can conclude that the use of proposed lightweight CNN can achieve better performance. Therefore, the good performance of the proposed detection system derives from both.

The developed model can be kept in the cloud for further processing. Using the developed system, the unknown recording to be detected is sent for remapping frog populations. Specifically, *Kroombit tinker frog* is first distinguished from the background noise. Then, averaged energy of detected calls can be used to estimate the number of *Kroombit tinker frog*.

6. Conclusion and future work

In this study, we develop a frog calling activity detecting system for continuous field recordings using CNN-GAP with multi-view spectrograms. We first compare the detection performance between single-view and multi-view spectrograms. In addition, we compare various multi-view spectrograms and lightweight models. Using the combination of multi-view spectrograms and lightweight models, our proposed method can achieve the highest detection accuracy and F1-score. To further improve the detection results, twin loss, a combination of BCE loss and focal loss is used. Finally, our proposed model can achieve the best performance using the fewest parameters.

Future work aims to propose a two-stage detection framework, where the first step is to detect frog calling activity and the second is to classify frog species. In addition, we will further investigate the fusion of 1D-, 2D-, and 3D-CNN models for improving frog calling activity detection using optimization methods such as swarm intelligence (Nayyar & Nguyen, 2018). Finally, since it is time-consuming to manually labeling the collected data, developing unsupervised methods to study bioacoustic data is in high demand.

CRediT authorship contribution statement

Jie Xie: Conceived and designed the analysis, Performed the analysis, Wrote the paper. **Mingying Zhu:** Performed the analysis, Wrote the paper. **Kai Hu:** Conceived and designed the analysis, Performed the analysis. **Jinglan Zhang:** Contributed data or analysis tools, Wrote the paper. **Harry Hines:** Collected the data, Contributed data or analysis tools Other contribution (Revise the paper). **Ya Guo:** Conceived and designed the analysis, Performed the analysis, Wrote the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant No: 61902154). This work is also partially supported by Natural Science Foundation of Jiangsu Province (Grant No: BK2019043526) and Jiangsu Province Post Doctoral Fund (Grant No: 2020Z430). This work is partially supported by the 111 Project. This work is also supported by Jiangsu Province Post Doctoral Fund (Grant No: 2020Z430) and China Postdoctoral Science special Foundation (Grant No: 2021T140281).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

Adavanne, S., Drossos, K., Çakir, E., & Virtanen, T. (2017). Stacked convolutional and recurrent neural networks for bird audio detection. In *2017 25th European signal processing conference (EUSIPCO)* (pp. 1729–1733). IEEE.

Al Bashit, A., & Valles, D. (2018). A mel-filterbank and MFCC-based neural network approach to train the Houston Toad call detection system design. In *2018 IEEE 9th annual information technology, electronics and mobile communication conference (IEMCON)* (pp. 438–443). IEEE.

Alzubi, J., Nayyar, A., & Kumar, A. (2018). Machine learning from theory to algorithms: an overview. In *Journal of physics: Conference series, Vol. 1142*. IOP Publishing, Article 012012.

Bedoya, C., Isaza, C., Daza, J. M., & López, J. D. (2014). Automatic recognition of anuran species based on syllable identification. *Ecological Informatics*, *24*, 200–209.

Brodie, S., Allen-Ankins, S., Towsey, M., Roe, P., & Schwarzkopf, L. (2020). Automated species identification of frog choruses in environmental recordings using acoustic indices. *Ecological Indicators*, *119*, Article 106852.

Caruso, F., Alonge, G., Bellia, G., De Domenico, E., Grammauta, R., Larosa, G., et al. (2017). Long-term monitoring of dolphin biosonar activity in deep pelagic waters of the Mediterranean Sea. *Scientific Reports*, *7*(1), 1–12.

Chen, W.-P., Chen, S.-S., Lin, C.-C., Chen, Y.-Z., & Lin, W.-C. (2012). Automatic recognition of frog calls using a multi-stage average spectrum. *Computers & Mathematics with Applications*, *64*(5), 1270–1281.

Cheng, Q., Li, H., Wu, Q., & Ngan, K. N. (2020). Hybrid-loss supervision for deep neural network. *Neurocomputing*, *388*, 78–89.

Chollet, F., et al. (2015). Keras. https://keras.io.

Colonna, J. G., Carvalho, J. R., & Rosso, O. A. (2020). Estimating ecoacoustic activity in the Amazon rainforest through Information Theory quantifiers. *PLoS One*, *15*(7), Article e0229425.

Colonna, J. G., Nakamura, E. F., & Rosso, O. A. (2018). Feature evaluation for unsupervised bioacoustic signal segmentation of anuran calls. *Expert Systems with Applications*, *106*, 107–120.

de Oliveira, A. G., Ventura, T. M., Ganchev, T. D., de Figueiredo, J. M., Jahn, O., Marques, M. I., et al. (2015). Bird acoustic activity detection based on morphological filtering of the spectrogram. *Applied Acoustics*, *98*, 34–42.

Driedger, J., Müller, M., & Disch, S. (2014). Extending harmonic-percussive separation of audio signals. In *ISMIR* (pp. 611–616).

Fitzgerald, D. (2010). Harmonic/percussive separation using median filtering. In *Proceedings of the international conference on digital audio effects (DAFx), Vol. 13*.

Gan, H., Zhang, J., Towsey, M., Truskinger, A., Stark, D., van Rensburg, B. J., et al. (2020). Data selection in frog chorusing recognition with acoustic indices. *Ecological Informatics*, *60*, Article 101160.

Himawan, I., Towsey, M., Law, B., & Roe, P. (2018). Deep learning techniques for koala activity detection. In *INTERSPEECH* (pp. 2107–2111).

Huang, C.-J., Yang, Y.-J., Yang, D.-X., & Chen, Y.-J. (2009). Frog classification using machine learning techniques. *Expert Systems with Applications*, *36*(2), 3737–3743.

Huzaifah, M. (2017). Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. arXiv preprint arXiv:1706.07156.

Jahn, O., Ganchev, T. D., Marques, M. I., & Schuchmann, K.-L. (2017). Automated sound recognition provides insights into the behavioral ecology of a tropical bird. *PLoS One*, *12*(1), Article e0169041.

Jain, A., & Nayyar, A. (2020). Machine learning and its applicability in networking. In *New age analytics* (pp. 57–79). Apple Academic Press.

LeBien, J., Zhong, M., Campos-Cerqueira, M., Velev, J. P., Dodhia, R., Ferres, J. L., et al. (2020). A pipeline for identification of bird and frog species in tropical soundscape recordings using a convolutional neural network. *Ecological Informatics*, *59*, Article 101113.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988).

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., et al. (2015). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, Vol. 8.

Nayyar, A., & Nguyen, N. G. (2018). Introduction to swarm intelligence. In *Advances in swarm intelligence for optimizing problems in computer science* (pp. 53–78). Chapman and Hall/CRC.

Pandeya, Y. R., Kim, D., & Lee, J. (2018). Domestic cat sound classification using learned features from deep neural nets. *Applied Sciences*, *8*(10), 1949.

Press, W. H., & Teukolsky, S. A. (1990). Savitzky-Golay smoothing filters. *Computers in Physics*, *4*(6), 669–672.

Tomasini, M., Smart, K., Menezes, R., Bush, M., & Ribeiro, E. (2017). Automated robust anuran classification by extracting elliptical feature pairs from audio spectrograms. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 2517–2521). IEEE.

Wimmer, J., Towsey, M., Planitz, B., Roe, P., & Williamson, I. (2010). Scaling acoustic data analysis through collaboration and automation. In *2010 IEEE sixth international conference on e-science* (pp. 308–315). IEEE.

Wimmer, J., Towsey, M., Roe, P., & Williamson, I. (2013). Sampling environmental acoustic recordings to determine bird species richness. *Ecological Applications*, *23*(6), 1419–1428.

Woodford, J. E., & Meyer, M. W. (2003). Impact of lakeshore development on green frog abundance. *Biological Conservation*, *110*(2), 277–284.

Xie, J., Hu, K., Hines, H., Zhang, J., Guo, Y., & Yu, J. (2020). Investigation of CNN-based models for frog calling activity detection. In *2020 IEEE international conference on signal processing, communications and computing (ICSPCC)* (pp. 1–4). IEEE.

Xie, J., Hu, K., Zhu, M., & Guo, Y. (2020). Bioacoustic signal classification in continuous recordings: syllable-segmentation vs. sliding-window. *Expert Systems with Applications*, Article 113390.

Xie, J., Hu, K., Zhu, M., Yu, J., & Zhu, Q. (2019). Investigation of different CNN-based models for improved bird sound classification. *IEEE Access*, *7*, 175353–175361.

Xie, J., Michael, T., Zhang, J., & Roe, P. (2016). Detecting frog calling activity based on acoustic event detection and multi-label learning. *Procedia Computer Science*, *80*, 627–638.

Xie, J., Towsey, M., Yasumiba, K., Zhang, J., & Roe, P. (2015). Detection of anuran calling activity in long field recordings for bio-acoustic monitoring. In *2015 IEEE tenth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)* (pp. 1–6). IEEE.

Xie, J., Towsey, M., Zhu, M., Zhang, J., & Roe, P. (2017). An intelligent system for estimating frog community calling activity and species richness. *Ecological Indicators*, *82*, 13–22.

Zhao, Z., Xu, Z.-y., Bellisario, K., Zeng, R.-w., Li, N., Zhou, W.-y., et al. (2019). How well do acoustic indices measure biodiversity? Computational experiments to determine effect of sound unit shape, vocalization intensity, and frequency of vocalization occurrence on performance of acoustic indices. *Ecological Indicators*, *107*, Article 105588.

Zhao, Z., Zhang, S.-h., Xu, Z.-y., Bellisario, K., Dai, N.-h., Omrani, H., et al. (2017). Automated bird acoustic event detection and robust species classification. *Ecological Informatics*, *39*, 99–108.