AUTOMATED ROBUST ANURAN CLASSIFICATION BY EXTRACTING ELLIPTICAL FEATURE PAIRS FROM AUDIO SPECTROGRAMS

Marcello Tomasini[1], Katrina Smart[1], Ronaldo Menezes[1], Mark Bush[2], and Eraldo Ribeiro[1]

[1]School of Computing
[2]Department of Biological Sciences
Florida Institute of Technology
Melbourne, Florida, USA

ABSTRACT

Ecologists can assess the health of wetlands by monitoring populations of animals such as Anurans (i.e., frogs and toads), which are sensitive to habitat changes. But, surveying anurans requires trained experts to identify species from the animals' mating calls. This identification task can be streamlined by automation. To this end, we propose an automatic frog-call classification algorithm and a smartphone application that drastically simplify the monitoring of anuran populations. We offer three main contributions. First, we introduce a classification method that has an average accuracy of 86% on a dataset of 736 calls from 48 anuran species from the United States. Our dataset is much larger and diverse than those of previous works on anuran classification. Second, we extract a new type of spectrogram feature that avoids syllable segmentation and the manual cleaning of the recordings. Our method also works with recordings of variable length. Third, our method uses GPS location and a voting scheme to reliably deal with a large number of species and high levels of noise.

***Index Terms*—** frog-call classification, machine learning, MSER, spectrograms, k-NN

1. INTRODUCTION

Scientists and governments grow increasingly concerned with the preservation of natural habitats. To preserve habitats, we must reliably track their health. This task can be especially challenging in wetlands because of the accessibility constraints of flooded areas. Fortunately, ecologists can assess wetlands' health by surveying populations of organisms whose own health suffer as a result of drastic habitat changes [1]. These organisms are called *bioindicators* and ones that are remarkably effective are anurans (i.e., frogs and toads) due to their permeable skin [2]. Anurans have helped scientists measure the impact of using chemicals in farming [3], and the impact of livestock on rivers [4].

When monitoring wetlands, ecologists identify anuran species by listening to hours of recorded mating calls. Surprisingly, this manual-identification task is common to even large-scale monitoring programs such as FrogWatch USA[1] and the North-American Amphibian Monitoring Program[2]. Such programs can largely benefit from automation.

Previous works on automated anuran identification used machine learning but differ mostly in the features used by classifiers. Huang et al. [5] was one the first to use machine learning to classify frog calls. Their approach extracts three types of features: spectral centroid, signal bandwidth, and threshold-crossing rate to train a k-NN classifier and a support vector machine (SVM) classifier to reach accuracy rates up to 89.05% and 90.30%, respectively. Acevedo et al. [6] compared the accuracy of linear discriminant analysis (71.45%), decision tree (89.20%), and SVM (94.95%) using features such as minimum and maximum frequencies, duration, and maximum power. Bedoya et al. [7] proposed an unsupervised classifier based on fuzzy rules and used the Mel-Frequency Cepstral Coefficients (MFCC) as features. Recently, Xie et al. [8] treated audio spectrograms as images from which they extracted features such as ridges and edges.

While the above methods achieved good progress towards the automation of frog classification, there is still room for improvement. Some approaches need syllable segmentation [5, 6, 7, 8]. Others need expensive pre-processing [6, 7, 8]. Some approaches were tested using small datasets, with few samples or few species from small geographic regions. For example, Huang et al. [5] used 25 calls from 5 species. Acevedo et al. [6] dataset had 9 frog and 3 bird species of Puerto Rico. Bedoya et al. [7]'s two datasets had 13 and 6 species from Colombia. Finally, some studies computed classification accuracy as the number of correctly classified syllables over the total number of syllables [5, 8], a practice that may result in testing the classifier on syllables from the same call.

In this paper, we introduce a new algorithm that uses image-processing techniques for classifying anuran species

This material is based upon work supported by the National Science Foundation under Grant No. 1152306.

[1]https://www.aza.org/frogwatch
[2]https://www.pwrc.usgs.gov/naamp/

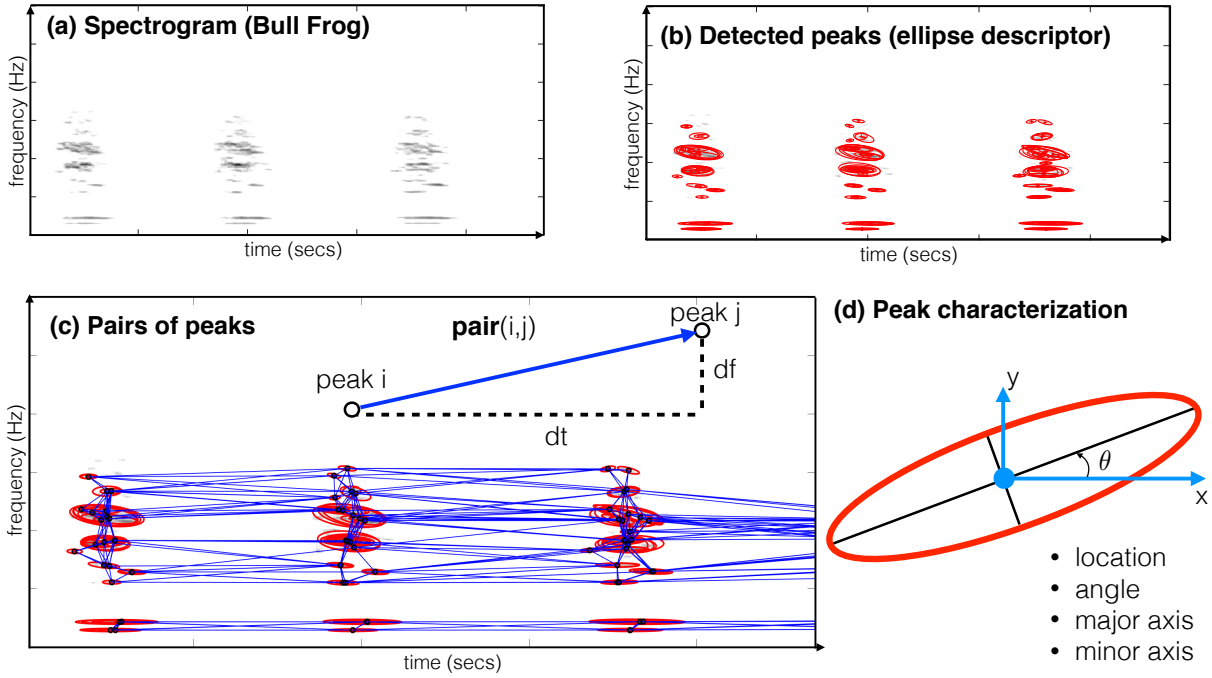

Fig. 1. Feature extraction overview. (a) Spectrogram, (b) Peak regions detected as MSER features, (c) Pairs of peaks, (d) Individual peak characterization.

from mating calls. The algorithm underpins the *WhatFrog* smartphone application[3]. The WhatFrog App can largely improve our ability to monitor wetlands while reducing the need for manual identification and expensive equipment (e.g., directional microphones). We tested our method on a dataset of 736 recordings of 48 species from the USA. The features we extract and the classification process are highly resilient to noise in both the training and the test data. Furthermore, our method does not need syllable segmentation, and it works with recordings of variable length. The method uses the phone's GPS to improve classification accuracy.

2. DATA AND METHODS

2.1. The dataset

The dataset provided by the Florida Tech's Paleoecology Laboratory has 736 frog calls from the 50 states of the USA, the District of Columbia, and the Virgin Islands. Georgia has the most species in the dataset (i.e., 26). Alaska, Hawaii, and the Virgin Islands have 2 species each. The average number of species per state is 13. For each species, we have from 10 to 23 samples, averaging 15 samples per species. Recordings last from one to 40 seconds, and contain ambient noise (e.g., water flowing, wind blowing, birds singing, people talking). Some recordings contain multiple species and choruses. Also, audio volume varies across samples.

[3]The WhatFrog App (`http://whatfrog.org`)

2.2. Pre-processing

Our method does not require noise cleaning. But, it helps improve identification for small training datasets. To train our classifier, we manually cleaned noise from each audio sample using the software ISSE [9].

Recordings were converted from stereo to mono, and then to spectrograms using a Short-Time Fourier Transform (STFT) with 90% overlap, window size of 2048, and maximum frequency of 9 KHz (i.e., includes all frog calls in the dataset). Spectrograms were thresholded by keeping only the 20% of their maximum magnitude (Figure 1a). Our method assumes that the anuran call is the loudest signal in the recording. Frequencies below 100 Hz were removed. We applied a 5×5 Gaussian filter with standard deviation of 2. Spectrogram values were normalized to be in the $[0, 1]$ range.

3. EXPERIMENTS

3.1. Feature extraction

The spectrograms produced by the pre-processing step are the input from which our method detects sets of high-energy peaks. These peaks are grouped into pairs of time-frequency features. The use of paired spectral peaks is also done by the music-matching method Shazam [10]. However, instead of spectrogram peaks, our method forms pairs of interest regions in the spectrogram. These regions are detected by the Maximally Stable Extremal Regions (MSER) algorithm [11],

Algorithm 1 Classification of an anuran call audio sample.

$Mdl \coloneqq$ selected classifier
$C \coloneqq$ labels predicted for F by Mdl
$P \coloneqq$ vector of class probabilities
repeat
  $X \leftarrow$ audio sample
  $S \leftarrow$ `createSpectrogram(`X`)`
  $F \leftarrow$ `extractFeatures(`S`)`
until `size(`F`) < threshold`
if `GPS` **then**
  $Mdl \leftarrow$ `getKNN(GPS)`
else
  $Mdl \leftarrow$ `generalKNN()`
end if
for $i = 1$ **to** $i = 8$ **do**
  $F_i = (F_i - \mu_i)/\sigma_i$
end for
$C \leftarrow Mdl(F)$
$P \leftarrow$ `normHist(`C`)`
$C_X \leftarrow \arg\max_{c \in C} P(c)$
return C_X

which describes each spectral region as an ellipse.

The steps of feature extraction are as follows. We scale down the thresholded spectrogram to one third of their original size. Then, we apply the MSER algorithm. The scaling step speeds up MSER calculation. The MSER algorithm describes regions as an ellipse (Figure 1b). Once the regions are at hand, we remove overlapping ellipses to reduce redundant features. Here, we by use non-maximum suppression with 50% as the overlap threshold parameter. From the remaining fitted ellipses, we extract three properties: (1) Frequency f_i, and time t_i which are the coordinates of the centroid of the ellipse that has the same second moment as the fitted region, (2) The orientation θ_i, which is the angle of the ellipse as measured from the horizontal direction to the ellipse's major axis (Figure 1d), and (3) Scale λ_i, which is the product of the ellipse's axes. The set $\mathbf{p}_i = \{f_i, t_i, \theta_i, \lambda_i\}$ characterizes a *peak* of interest.

In the next step, our method creates pairs of peaks using a sliding window of size $dt \times 2df$ (Figure 1c). As the window slides along the time axis of the spectrogram, we connect each peak i up to the N closest peaks j such that $t_j \in [t_i, t_i + \mathtt{dt}]$ and $f_j \in [f_i - df, f_i + df]$. Value N is the fan-out factor, used to limit the number of pairs, and dt, df are the lookup sizes for the time and the frequency, respectively. In our tests, we set $N = 6$, $dt = 331$, and $df = 331$. Higher N increases noise robustness but also increases computational cost.

From each pair $(i, j), i < j$, we compute the time difference $\Delta t = t_j - t_i$ and the frequency difference $\Delta f = f_j - f_i$. The final feature vector is:

$$\mathbf{f} = (f_i, \lambda_i, \theta_i, f_j, \lambda_j, \theta_j, \Delta t, \Delta f)\,. \tag{1}$$

Better classification accuracy results from features that have enough entropy to minimize spurious matches. By using pairs of peaks instead of individual peaks, we increase the amount of information encoded in each feature.

3.2. Classification algorithm

The classification method consists of a set of k-Nearest-Neighbors (k-NN) classifiers, each trained for frogs from each U.S. state. The corresponding classifier is selected using the smartphone's GPS coordinates. If GPS location is unavailable, the method falls back on a k-NN classifier that has been trained for all frog species. In both types of classifiers, we used $k = 1$. Once the classifier is selected, each feature i of feature vectors $\mathbf{f}$ is standardized according to the mean μ_i and standard deviation σ_i computed from the training data of the state. This normalization step is key to improve k-NN's classification accuracy. We use a voting scheme where the state-specific classifier predicts the class c of each feature vector. We then compute the frequency of each class and we normalize it to produce a vector of probabilities. The predicted class is the one with the highest probability. This voting scheme makes the algorithm robust to noise, even when several feature vectors are miss-classified. Moreover, the longer the anuran call, the higher the classification accuracy. The algorithm also outputs the class probabilities. These probabilities can be useful in the future to help us automatically select samples to use for online machine learning and to provide a confidence measure in the classification accuracy.

We trained and tested the classifiers using a 10-fold cross validation (CV) scheme. The training samples were in two formats: *clean* (i.e., manually cleaned recordings) or *noisy* (i.e., original recordings). The noisy samples were used for testing the classification method. Here, we tried to simulate a more realistic recording situation. For the GPS-enabled classification, we ran a 10-fold cross validation per state/region (i.e., a total of 52 classifiers). The average classification accuracy was computed as the average of the classification accuracy of the states/regions. We also provide the minimum classification accuracy as the worst classification accuracy out of all regions. For the no-GPS classifier, the average accuracy was the average accuracy of the 10-cross-validation runs.

Our algorithm presented good degree of noise robustness during training. There was little difference between clean and noisy training data when the GPS was enabled (Figure 2). Classification accuracy actually *increased* in the case of the single classifier (i.e., no GPS) due to the larger number of feature vectors extracted from the noisy data. Furthermore, classification accuracy increased up to 3% by requiring a higher minimum number of detected features to accept the sample as a good sample. We tested the method with no threshold, 50 features, and 100 features. For comparison, from a 30-second long sample there can be hundreds of feature vectors

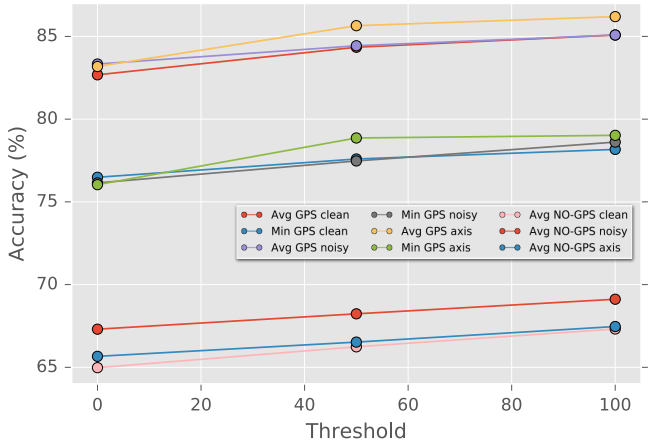

Fig. 2. Classification accuracy as function of the number of features per sample. Using more features improves accuracy.

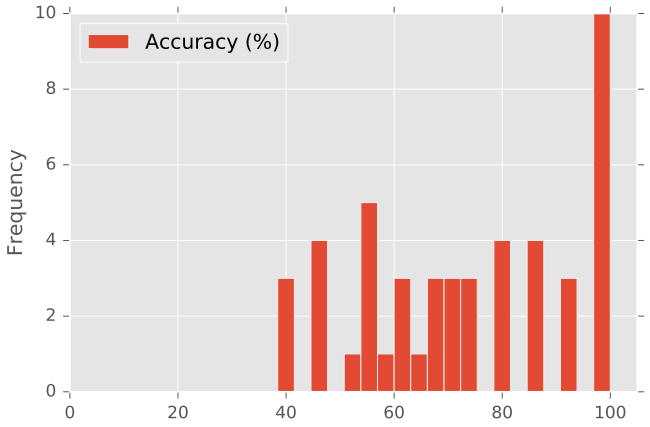

Fig. 3. Accuracy distribution when no GPS is available. Some species are still hard to identify.

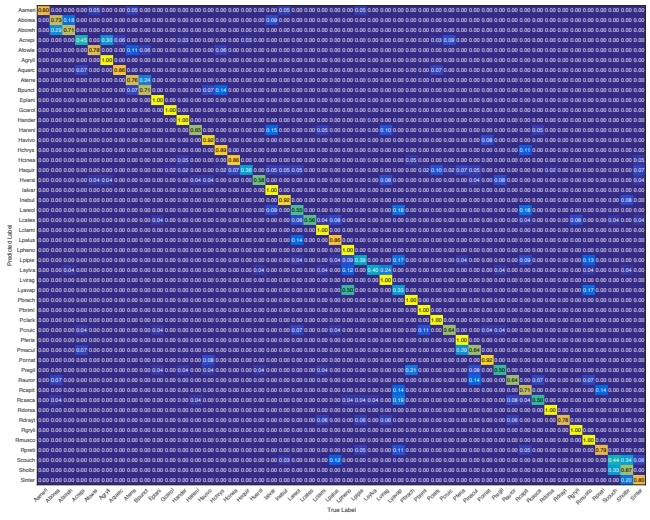

Fig. 4. Confusion matrix resulted from classification using kNN trained for all 48 species (i.e., No GPS).

extracted, often more than 500. The algorithm also provides a trade-off parameter to balance the length of recorded sample (i.e., computational cost) and resilience to noise and classification accuracy (i.e., classification performance).

The multi-classifier approach (i.e., GPS) reached a minimum accuracy of 78.61% and an average accuracy of 85.09%, greatly outperforming the single-classifier approach (i.e., no GPS) which had an accuracy of up to 67.31% on clean training data and up to 69.11% on noisy training data. In fact, we observed a strong correlation between the state-based classification accuracy and the number of species per state, with a Pearson correlation coefficient $r \in [0.65, 0.75]$. That is, the higher the number of species, the harder was to classify the samples. Furthermore, when using the single classifier, some species were particularly hard to classify: while more than 20 species had accuracy greater than 80%, 6 species were correctly classified less than 50% of the time (Figure 3).

We also tested a different type of feature vector, where the scale λ is replaced by the axes of the ellipse. While scale is correlated to the axes' lengths, features can have the same scale but different axes. By using the axes lengths, the feature vectors showed increased discriminative power. However, accuracy was improved by $\sim 1\%$, reaching up to 86.19%, only for the classifier with GPS availability, but not for the classifier trained on all the 48 species (Figure 2).

A confusion matrix with classification results for all frog species is shown in Figure 4.

4. CONCLUSION AND FUTURE WORK

We introduced a new method for automated anuran-call classification. Our method's main contribution is to characterize regions around spectral peaks as elliptical shapes. These regions are then grouped together into pairs to form the features used by a voting-based k-NN classifier. Tests done on a large dataset of frog calls demonstrated our method's effectiveness. The method does not suffer of any of the limitations of previous methods, i.e., it does not require syllable segmentation, extensive call cleaning, or any other heavy pre-processing step. The main feature-extraction steps are to covert the audio call to a spectrogram representation. The method is quite robust to noise in training and test data, and it allows audio samples to be of variable length.

As for future work, we need to identify the reasons why certain calls are hard to classify. We also want to investigate other ways to represent pairs and other structural configurations of peaks to try to improve classification accuracy.

5. REFERENCES

[1] John F McCarthy and Lee R Shugart, "Biomarkers of environmental contamination," 1990.

[2] Abhishek D Garg and Rajshekhar V Hippargi, "Significance of frogs and toads in environmental conservation," 2007.

[3] Ross V Hyne, S Wilson, and Maria Byrne, "Frogs as bioindicators of chemical usage and farm practices in an irrigated agricultural area," *Final Report to Land & Water Australia*, 2009.

[4] Amy Jansen and Michael Healey, "Frog communities and wetland condition: relationships with grazing by domestic livestock along an australian floodplain river," *Biol. Conservation*, vol. 109, no. 2, pp. 207–219, 2003.

[5] Chenn-Jung Huang, Yi-Ju Yang, Dian-Xiu Yang, and You-Jia Chen, "Frog classification using machine learning techniques," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3737–3743, 2009.

[6] Miguel A Acevedo, Carlos J Corrada-Bravo, Héctor Corrada-Bravo, Luis J Villanueva-Rivera, and T Mitchell Aide, "Automated classification of bird and amphibian calls using machine learning: A comparison of methods," *Ecological Informatics*, vol. 4, no. 4, pp. 206–214, 2009.

[7] Carol Bedoya, Claudia Isaza, Juan M Daza, and José D López, "Automatic recognition of anuran species based on syllable identification," *Ecological Informatics*, vol. 24, pp. 200–209, 2014.

[8] Jie Xie, Michael Towsey, Jinglan Zhang, Xueyan Dong, and Paul Roe, "Application of image processing techniques for frog call classification," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4190–4194.

[9] Nicholas J Bryan, Gautham J Mysore, and Ge Wang, "Isse: an interactive source separation editor," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2014, pp. 257–266.

[10] Avery Wang et al., "An industrial strength audio search algorithm.," in *ISMIR*. Washington, DC, 2003, pp. 7–13.

[11] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.