ANURAN CALL CLASSIFICATION WITH DEEP LEARNING

Julia Strout1, Bryce Rogan2, S.M. Mahdi Seyednezhad3, Katrina Smart3, Mark Bush4, and Eraldo Ribeiro3

1Department of Computer Science, University of South Carolina
2Department of Mathematics, Pomona College
3School of Computing, Florida Institute of Technology
4Department of Biological Sciences, Florida Institute of Technology

ABSTRACT

Ecologists can assess the health of flooded habitats or wetlands by studying the variations in the populations of bioindicators such as anurans (i.e., frogs and toads). To monitor anuran populations, ecologists manually identify anuran species from audio recordings. This identification task can be significantly streamlined by the availability of an automated method for anuran identification. Previous promising frog-call identification methods have relied on the extraction of pre-designed features from audio spectrograms such as Mel Coefficients and other filter responses. Instead of using pre-designed features, we propose to allow a deep-learning algorithm to find the features that are most important for classification. In work reported in this paper, we used two deep-learning methods that apply convolutional neural networks (CNN) to anuran classification. Transfer learning was also used. The CNN methods was tested on our dataset of 15 frog species, and produced a classification accuracy up to 77%.

Index Terms— Deep learning, transfer learning, anuran call, sound classification.

1. INTRODUCTION

Societies are increasingly concerned about the conservation of natural habitats, especially those with high-degree of diversity such as wetlands and floodplains. To monitor the health of wetlands, ecologists can use the help of bioindicators. These are organisms whose health reflects the health of their habitat. An effective bioindicator is the family of frogs and toads, or anurans [1]. To study changes in anuran populations, ecologists identifying species from these animals’ mating calls in the field or from audio recordings. This manual monitoring approach is sometimes done by trained volunteers of large-scale programs such as FrogWatch USA and the North-American Amphibian Monitoring Program (NAMP). However, even trained professionals can take hours to classify a large batch of recordings. Thus, the automation of the identification of anurans from calls can improve data collection in wetlands conservation research.


Once syllables are segmented, classification methods extract features that are used for training and classification. Feature-extraction methods aim at obtaining the most distinctive features of the call, and the choice of features an have a large impact on the quality of results. Features that have been used in previous anuran-call classification methods include the Mel-frequency cepstral coefficients (MFCCs) used by Bedoya et al. and spectral centroids, signal bandwidth, and threshold-crossing rate used by Huang et al. [4, 5]. Xie et al. also used MFCCs, along with acoustic event detection and ridge detection, while Acevedo et al. created a feature vector from call duration, maximum power, and minimum and maximum frequencies [3, 2].

Designing features for classification is nontrivial. Ideally, features should be learned from training data. This is the idea underlying deep-learning methods [6], which are implemented by convolutional neural networks Convolutional neural networks (CNNs) [7], with a primary focus on classifying image data. CNNs have also been applied to sound classification by converting audio data to spectrograms. For example, Ossama et al. applied CNN to speech recognition [8]. More recently, Sanaith et al. [9] found that CNNs performed better than standard deep neural networks on a large vocabulary for continuous speech recognition (LVCSR).

In this paper, we use deep-learning methods for classifying frog calls. We tested two implementations of deep learning: R-CNN [10], AlexNet [11], and CaffeNet [12]. Our test dataset contains 212 calls from 15 species of frogs from the U.S.A. The content of our dataset is summarized in Ta-
Table 1: Anuran Call Data Set

<table>
<thead>
<tr>
<th>Species Name</th>
<th>Common Name</th>
<th>Acronym</th>
<th>Total Calls</th>
<th>Reduced Dataset Syllables</th>
<th>Cleaned Dataset Syllables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lithobates catesbeianus</td>
<td>Bull Frog</td>
<td>BF</td>
<td>17</td>
<td>87</td>
<td>112</td>
</tr>
<tr>
<td>Hyla gratiosa</td>
<td>Barking Tree Frog</td>
<td>BTF</td>
<td>9</td>
<td>70</td>
<td>82</td>
</tr>
<tr>
<td>Pseudacris nigrita</td>
<td>Southern Chorus Frog</td>
<td>CF</td>
<td>11</td>
<td>140</td>
<td>427</td>
</tr>
<tr>
<td>Gastrophrynus carolinensis</td>
<td>Eastern Narrow Mouth Toad</td>
<td>ENMT</td>
<td>9</td>
<td>45</td>
<td>43</td>
</tr>
<tr>
<td>Lithobates clamitans</td>
<td>Green Frog</td>
<td>GF</td>
<td>15</td>
<td>148</td>
<td>130</td>
</tr>
<tr>
<td>Hyla cinerea</td>
<td>Green Tree Frog</td>
<td>GTF</td>
<td>23</td>
<td>429</td>
<td>715</td>
</tr>
<tr>
<td>Pseudacris ocularis</td>
<td>Little Grass Frog</td>
<td>LGF</td>
<td>11</td>
<td>108</td>
<td>108</td>
</tr>
<tr>
<td>Anaxyrus quercicus</td>
<td>Oak Toad</td>
<td>OT</td>
<td>7</td>
<td>87</td>
<td>99</td>
</tr>
<tr>
<td>Rana grylio</td>
<td>Pig Frog</td>
<td>PF</td>
<td>16</td>
<td>76</td>
<td>73</td>
</tr>
<tr>
<td>Hyla femoralis</td>
<td>Pine Woods Tree Frog</td>
<td>PWTF</td>
<td>8</td>
<td>208</td>
<td>433</td>
</tr>
<tr>
<td>Acris gryllus</td>
<td>Southern Cricket Frog</td>
<td>SCF</td>
<td>21</td>
<td>291</td>
<td>236</td>
</tr>
<tr>
<td>Hyla squirella</td>
<td>Squirrel Tree Frog</td>
<td>SF</td>
<td>8</td>
<td>114</td>
<td>147</td>
</tr>
<tr>
<td>Lithobates phenoecephalus</td>
<td>Southern Leopard Frog</td>
<td>SLF</td>
<td>16</td>
<td>226</td>
<td>400</td>
</tr>
<tr>
<td>Pseudacris crucifer</td>
<td>Spring Peeper</td>
<td>SP</td>
<td>22</td>
<td>324</td>
<td>621</td>
</tr>
<tr>
<td>Anaxyrus terrestris</td>
<td>Southern Toad</td>
<td>ST</td>
<td>19</td>
<td>20</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 1. The audio recordings came from a mixture of web-based collections and field recording made in Florida by our team. Because our dataset is not as large as usually required by deep-learning techniques, we used transfer learning, i.e., a pretrained network as a feature extractor. Transfer learning has been shown to produce consistently higher classification rates than networks trained for specific tasks [13].

2. METHODS AND EXPERIMENTS

2.1. Data Preparation

Call distribution in our dataset is uneven across species, with some having as many as 20 calls and others with as few as 7. This amount of data is insufficient to train CNNs. Also, using full calls as input produces large spectrograms for the network to process. We compensated for the lack of data, and simultaneously reduced the input size, in two ways.

The first method we used to expand our data was to break the large spectrograms into smaller, sequential overlapping windows of $140 \times 200$ pixels. The original spectrogram image was $900 \times 713$ pixels. We then trained a MatConvNet architecture, which produced classification accuracies ranging from 53.02% to 61.48% depending on the window and overlap sizes. All results showed a great deal of overfitting (Figure 1a). The disparity between the blue lines, representing the accuracy of running the validation set through the network, and the black lines, representing the accuracy of running the training data, shows that while the network adjusted the weights to almost perfectly classified the training data, this accuracy did not extend to unseen data. However, the network often learned to classify these data before becoming general enough to handle new data, resulting in overfitting. The confusion matrix in Figure 1b offers insight into which species are being misclassified the most. From the matrix, we can see that the barking tree frog (BTF) is almost completely misclassified, with only a .08% accuracy. This frog is being confused with the bull frog (BF) about a third of the time and with the green tree frog (GTF) another third. Four species were perfectly classified with accuracies of 100% and corresponding yellow squares on the diagonal.

Our next step was to use a pre-trained network without retraining it, and instead just use it as a feature extractor by removing the vectors from one of the final fully connected layers. We then took these vectors as features and fed them to a support vector machine (SVM). This process yielded our best results by far. We used multiple different trained nets, each trained on the ImageNet database, but with different architectures. Within the nets we used, R-CNN, AlexNet, and CaffeNet, we extracted the feature vectors at different fully connected layers within the network. For each implementation, and each selected extraction layer, we cross validated our results with ten different training and testing data sets. Table 2 shows accuracy results from using a CNN as a feature extractor before training an SVM.

R-CNN is a model that uses high-capacity CNNs and bottom-up region proposals in order to localize and segment objects. We extracted features from multiple fully connected layers in the network. The best result, from layer fc7, offered a mean classification rate of 73.57%, a median of 73.89%, and a max of 81.88%. We introduced a voting mechanism to classify calls in their totality by taking the classification for each syllable within a call and then determining the classification for the entire call by choosing the species with the most votes. Using this method and the features extracted from layer fc-rcnn we got a max classification of 88.45% with a mean of 76.61%, and median of 76.84%.

AlexNet was trained on the 1.3 million images in the LSVRC-2010 ImageNet training set and consists of five convolutional layers, two fully connected layers, and a final soft-
max layer. When features were extracted from layer fc8, the SVM resulted in correct classification a mean of 70.72%, a median of 68.07%, and a max of 84.70%. The voting method significantly boosted these results to a mean of 78.39%, a median of 78.20%, and a max of 98.76%, our highest single result from any of the networks.

The final network we used as a feature extractor was CaffeNet [12]. CaffeNet has the same architecture as AlexNet except it was trained with data that was augmented differently and the pooling and normalization layers were switched. With this network, the results of the SVM were the highest with a mean of 76.97%, a median of 77.58%, and a max of 90.57%. The voting method increased accuracy to a mean of 80.37%, a median of 82.85%, and a max of 97.00%.

3. CONCLUSION

We tested the use of convolutional neural networks (CNN) for frog-call classification. We compared different CNN implementations by testing them on a dataset of about 200 calls from 15 frog species. We also used transfer learning. Results were promising and suggest that CNNs might offer a solution to the identification of frog species from calls.
4. REFERENCES


