IEEE International Conference on Acoustic, Speech, and Signal Processing - ICASSP 2017

ANURAN CALL CLASSIFICATION WITH DEEP LEARNING

Julia Strout[1], Bryce Rogan[2], S.M. Mahdi Seyednezhad[3], Katrina Smart[3], Mark Bush[4], and Eraldo Ribeiro[3]

[1]Department of Computer Science, University of South Carolina
[2]Department of Mathematics, Pomona College
[3]School of Computing, Florida Institute of Technology
[4]Department 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' matting 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.

Most anuran-call classification algorithms rely on syllable segmentation as a first step. Syllable variation contain identifying information about frog species. Acevedo et al. [2] extracted syllabus manually. Xie et al. [3] segmented syllabus automatically by combining acoustic event detection, Wiener filters, and Gaussian kernels. Bedoya et al. [4] and Huang et al. [5] also used automatic segmentation. Bedoya et al. used an energy thresholding algorithm, while Huang et al. used an iterative time-domain algorithm.

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-

This work was supported by the NSF grants No. 1560345 and No. 1152306

Table 1: Anuran Call Data Set

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

ble 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

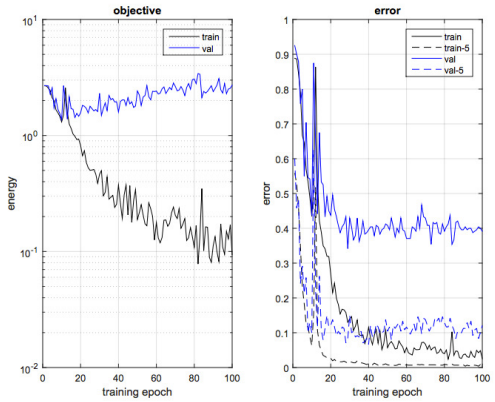
Call distribution in our dataset is uneven across species, with some having as many as 20 calls and others with as few as 7. This amount of data is insufficient to train CNNs. Also, using full calls as input produces large spectrograms for the network to process. We compensated for the lack of data, 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 140×200 pixels. The original spectrogram image was 900×713 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.

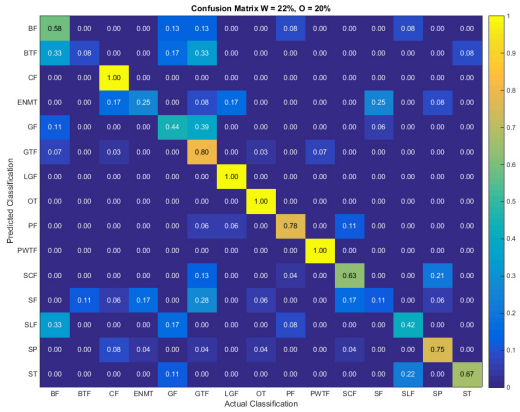
Our next step was to use a pre-trained network without retraining it, and instead just use it as a feature extractor by removing the vectors from one of the final fully connected layers. We then took these vectors as features and fed them to a support vector machine (SVM). This process yielded our best 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.[10] 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-

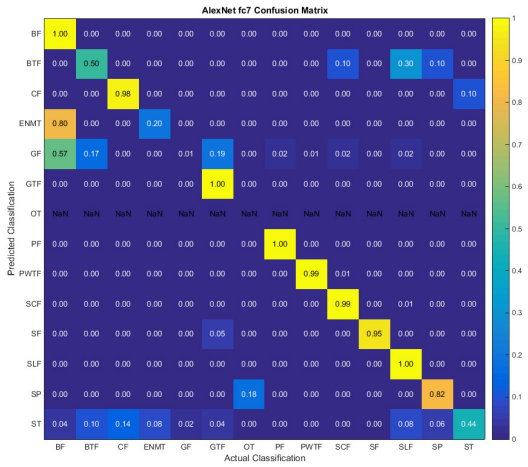


(a) Energy and Error Graph

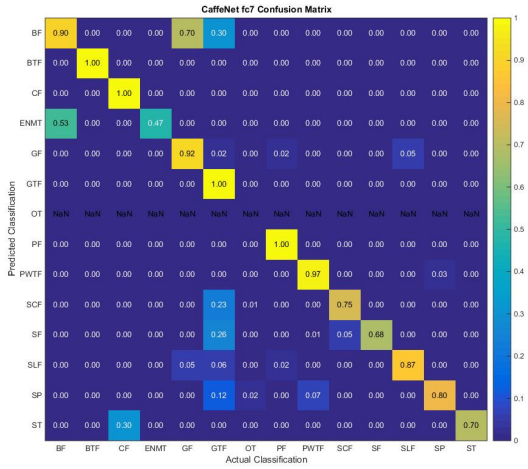


(b) Confusion Matrix

Fig. 1: Results from windows of 200 pixels and overlap of 40 pixels. Overall classification score of 60.83%.



(a) AlexNet normalized confusion matrix.



(b) CaffeNet normalized confusion matrix.

Fig. 2: Results from two different architectures with the hybrid CNN-SVM on the reduced dataset of 159 calls.

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%.

Table 2: Full accuracy results from using a CNN as a feature extractor before training an SVM

Network and Layer	Mean without Voting	Median without Voting	Maximum without Voting	Mean with Voting	Median with Voting	Maximum with Voting
R-CNN, fc6	73.07%	71.84%	81.78%	79.29%	79.86%	**85.24%**
R-CNN, fc7	73.57%	73.89%	81.88%	77.27%	76.84%	**82.02%**
R-CNN, fc-rcnn	67.89%	68.57%	77.02%	76.61%	77.37%	**88.45%**
AlexNet, fc6	44.51%	43.65%	62.49%	44.74%	41.87%	**67.32%**
AlexNet, fc7	71.64%	71.01%	85.51%	73.51%	73.20%	**90.26%**
AlexNet, fc8	70.72%	68.07%	84.70%	78.39%	78.20%	**98.76%**
CaffeNet, fc6	54.10%	51.35%	66.86%	55.55%	55.91%	**76.39%**
CaffeNet, fc7	76.97%	77.58%	90.56%	80.37%	82.85%	**97.00%**
CaffeNet, fc8	75.66%	76.23%	82.06%	79.31%	79.64%	**84.85%**

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

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

[2] 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.

[3] 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.

[4] 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.

[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] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[7] Yann LeCun and Yoshua Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 1995, 1995.

[8] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, and Gerald Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4277–4280.

[9] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8614–8618.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition*, 2014.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[13] Ali Sharif Razavian, Hossein Azizpour, and Josephine Sullivan andStefan Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," *CoRR*, vol. abs/1403.6382, 2014.