Speaker Detection ... for Birds

Eli Pugh
Stanford University
epugh@stanford.edu

Abstract

While bioacoustic event detection has made significant progress in the past few years, one major challenge of even the best systems is adapting to new classes/data. In this paper, I experiment with different types of Prototypical Networks [1] for few-shot detection of bird calls as well as other bioacoustic detection datasets. I show that dilated convolutional neural networks (dCNNs) are a top performer for this task, while larger Long Short Term Memory (LSTM) and Transformer [2] networks struggle in the few-shot setting.

1 Introduction

Ecologists currently monitor bird migration patterns using weather radar tools that give insights into the speed, density, and direction of bird movements. These tools aren't able to aren't able to distinguish what species are migrating, and rely on human observations that are largely absent at night (when most migrate).

By using a network of microphones identifying birds by their vocalizations, one could monitor timings for different species much more accurately.

I hope to improve upon current best models in bird call identification and detection to create an application that could be used to more accurately track bird migration patterns. A system that adapted to new species with only a few data samples would be very valuable to ecologists, especially given the lack of data for training in a typical supervised fashion.

2 Related Works

2.1 Bird Vocalization Detection/Classification

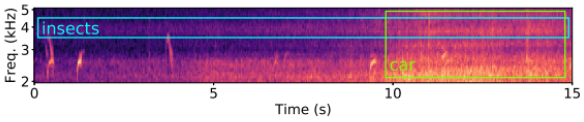
Recently there has been growing interest in bird call detection and classification, particularly in the BirdVox community. BirdVox is a collaboration between the Cornell Lab of Ornithology and NYU's Music and Audio Research Laboratory. The latest research efforts towards this goal have been compiled here, and there is a nice linear progression throughout time compared to some ML subfields, where it seems that progress is distributed in many directions / aspects of a task. Because of this, I'll do a fairly sequential review of recent progress.

In 2016, three of the first larger-scale bird vocalization datasets were released [3]. CLO-43SD is a classification dataset across 43 species. CLO-WTSP and CLO-SWTH are datasets for vocalization detection (for White-Throated Sparrow and Swainson's Thrush). They also demonstrate a model which uses unsupervised representation learning (patching and principal component analysis), and then trains a classifier on these smaller features (support vector machine).

In 2017, [4] improved upon their previous efforts with a convolutional model. They use a similar unsupervised representation learning methods (spherical K-means [5]) as well as convolutional neural network (CNN) and fuse them together late in the network for improved accuracy.

Per-Channel Energy Normalization (PCEN) [6][7] was first introduced in 2017, and soon became a big boost in nearly all systems going forward. This preprocessing method is designed for bioacoustic event detection, as it transforms the noise to look more Gaussian and helps events stand out. PCEN preprocessing is an adaption of the mel-frequency spectrogram, but smooths noise to look more Gaussian. This allows interesting events such as bird chirps to stand out. PCEN is shown in more detail in section 3.1.

Popular Software package BirdVoxDetect was released in 2019 [8] building off of previous methods for the goal of vocalization detection. They use a context-adaptive neural network (CANN) [9] layer as output to predictions. The CANN trains a network to adaptively produce the final fully-



(a) Logarithmic transformation.

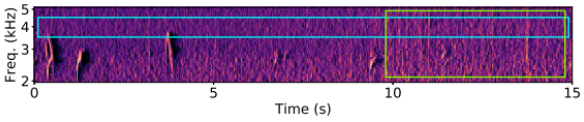


Figure 1: Log Mel vs PCEN

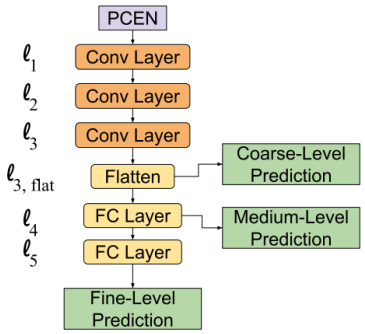


Figure 2: TaxoNet

connected output layer weights based on 60 second and 30 minute contexts for inference. They also use extensive data augmentation, including pitch shifting, time warping, and background noise augmentations using longer outdoor sound datasets.

Partner software package BirdVoxClassify was released in 2020 [10], and uses hierarchical classifier to not only improve classification accuracy, but improve closeness of missed examples. Taxonet builds on previous convolutional approaches, but also attempts to predict animal order, family, and species, with less specific predictions coming from hidden layers earlier in the network.

2.2 Few-Shot Event Detection

While BirdVoxDetect and BirdVoxClassify bring strong detection and classification software to the community as open source repositories, there has been little effort to extend into few-shot settings.

While there haven't been many efforts to build few-shot bird vocalization detection systems, there has been some recent research into few-shot keyword spotting, where we hope to detect events where a keyword is spoken only given a few speech samples of that keyword. This is a very similar task to few-shot bird call detection, and thus some study of architectures used would be very helpful when approaching this task.

[11] examine many common few-shot learning methods on audio data, such as nearest neighbor MFCCS, Prototypical Networks [1], and pretrained automatic speech recognition (ASR) models with fine-tuned softmax layer. This is the first real attempt to apply different popular few-shot learning techniques to audio classifiers and compare results. They show that ProtoNets achieve stellar results, though can sometimes be improved upon by pretrained models with finetuning depending on the number of support samples.

More recently, [12] show that temporal and dilated convolutions work well as embedding layers in prototypical networks for keyword spotting. They also compare other network structures to create an embedding space, such as a traditional small CNN, the original Protonet CNN, and ResNet8 [13].

3 Methodology

3.1 Featurization

Since most competitive bioacoustic event detection networks use Per-Channel Energy Normalization (PCEN) [6][7] as an acoustic model, I follow suit. PCEN is calculated by first taking the log mel-spectrogram $\mathbf{E}(t, f)$ at time t and frequency f, and transforming to Gaussianize noise. Below α, ϵ, r, and δ are positive constants.

$$\mathbf{PCEN}(t, f) = \left(\frac{\mathbf{E}(t, f)}{(\epsilon + (\mathbf{E} \star \theta_T)(t, f))^{\alpha}} + \delta \right)^{r} - \delta^{r}$$

Since this has the effect of smoothing noise and visually accentuating events in the spectrogram, it's a great preprocessing method to try and weed out natural background noise. Since most birds migrate at night, and there is often lots of ambient bug noise at night, it's important to smooth low-volume noise and highlight possible bird vocalizations.

An example of how PCEN transforms the log mel-spectrogram is given in Figure 1.

3.2 Prototypical Networks

ProtoNets are a simple yet effective approach to few-shot learning tasks, where we are given data for training a network, but on test time the classes are new, and we only see a few examples from each class to adjust the model before evaluating. Call training data D_{train} and testing data D_{test}.

Episodic training is used, where in each episode, the model is fed K "support" examples from each of N classes randomly sampled from the training

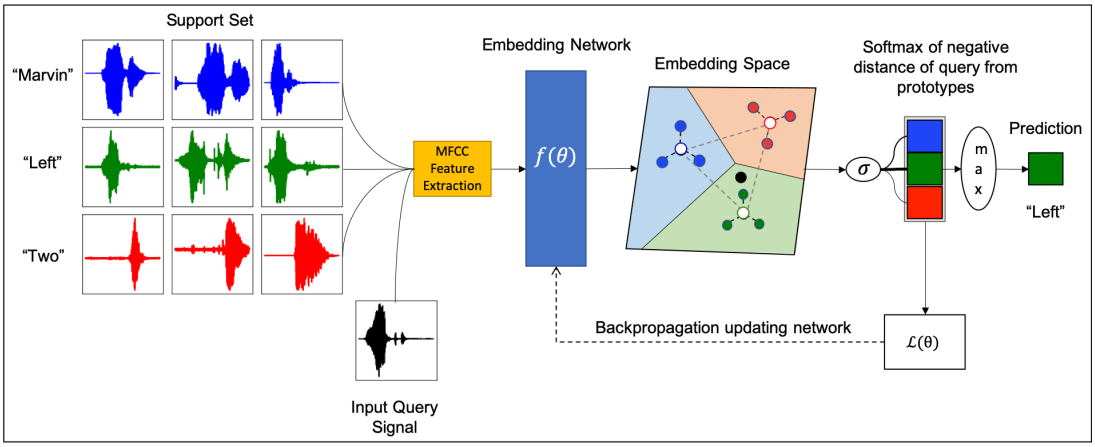


Figure 3: Prototypical Network Training (Image from [12])

set. These can be leveraged to adjust the model, and then the model performs classification on Q different "query" examples, also from D_{train}. This same episodic training style is used for testing as well, but where the classes in D_{test} are distinct from those in D_{train}.

ProtoNets use loss

$$L(\theta) = -\sum_{t=1}^{|Q_e|} \log P_\theta(y_t|q_t, S_e), \qquad (1)$$

where $(q_t, y_t) \in Q_e$ (query set for episode e), and S_e is the support set for episode e.

Prototypical networks pass each example of the support set through an embedding network to place them in a vector space, with the aim of having distances in the vector space represent similarity in the task. A centroid (mean) of each class's support set vectors is taken in the embedding space via averaging:

$$p_c = \frac{1}{|S_e^c|} \sum_{(s_i,y_i)\in S_e^c} f(s_i), \qquad (2)$$

where S_e^c is the support set for episode e class c, and f is our embedding network.

For inference, the query examples are fed through the same embedding network, and classified as the class with centroid having shortest Euclidean distance d to the query example. This is computed as a probability distribution by taking the softmax of negative distances:

$$P(y = c|q_t, S_e, \theta) = \frac{exp\Big(-d\Big(f(q_t), p_c\Big)\Big)}{\sum_n exp\Big(-d\Big(f(q_t), p_c\Big)\Big)} \qquad (3)$$

The network parameters θ using a favorite version of gradient descent (in this case, Adam [**?**]).

The name "Prototypical Network" would suggest that this is a network architecture, but ProtoNets are more of a training paradigm/regime/structure. The network itself can be chosen in any way that embeds examples into a vector space where the Euclidean distance metric separates classes effectively.

4 Experiments

Baseline: The baseline is one standard in bioacoustics: Normalized cross-correlation template matching [14]. This is a standard measure of correlation to spectrogram images in the support set. The closest one's class is chosen as the prediction.

4.1 ProtoNet Embedding Networks

CNN: The first ProtoNet used was a CNN as described in [12]. This network had 4 convolutional blocks, where each block consisted of a convolutional layer, batch normalization, ReLU activation, and a 2d max-pool. For the base setup, convolutional filters were 3×3 with stride 1, and max-pooling was 2×2. All 4 convolutional blocks had 128 channels.

Dilated CNN: With inspiration from architectures like WaveNet [15] using dilated convolutions to have wider temporal span for each dimension of the output, I decided to modify the existing network to have dilation 2, thus the kernel was then 3×3, but spaced out over an 8×8 grid.

Pyramidal LSTM: With inspiration from the Listen Attend Spell encoder [16], I also ran experiments with a pyramidal LSTM for creating the embeddings. This network is a traditional LSTM,

except at each layer, consecutive pairs of inputs are concatenated together. This means that each layer has sequence length half that of the previous layer. For experiments, hidden dimension 128, bidirectional LSTM cells are used, and there are 4 pyramidal layers.

Transformer: The transformer network structure is the encoder originally seen in [2]. Traditional sine/cosine positional encoding was used, and each block is multi-head attention with a residual connection, then two feed-forward layers with a residual connection. The model used had hidden dimension 128, 16 heads per block, and 8 blocks.

4.2 Data

All data is sourced from the 2021 Detection and Classification of Acoustic Scenes and Events (DCASE) Few-Shot Bioacoustic Event Detection Challenge. Data and challenge information can be found here: http://dcase.community/challenge2021/task-few-shot-bioacoustic-event-detection.

This challenge uses data from BirdVox-DCASE-10h, a dataset collected from recordings in Tompkins County, New York, United States. There are 11 bird classes, and a total of 10 hours of data. Though there are 10 hours of data, there are only 2662 events, and thus there is a massive imbalance of noise to events.

The challenge also compiles and uses data from spotted heyena recordings, jackdaw recordings, and meerkat recordings. I train on and use these recordings for evaluation as well, though most of the data (and my focus for this project) is on bird vocalizations.

4.3 Results

While all networks performed significantly better than baseline, there was a surprisingly large discrepancy in performance depending on the embedding network used.

ProtoNet CNN with dilations was the clear winner, while also remaining very fast. The transformer embedding model worked well, but was incredibly painful to train, and required lots of fiddling to learn at all. In addition, it was painfully slow during both training and inference. It's unclear why the LSTM struggled, but maybe it's difficult to pass gradients through with such few samples and over a relatively finnicky episodic training regime.

System	F-Measure	Precision	Recall
Template Matching Baseline	2.01	1.08%	14.46%
ProtoNet CNN	35.10	**52.61%**	26.33%
ProtoNet Deep CNN+Dilations	**42.25**	48.25%	**37.58%**
ProtoNet Pyramidal LSTM	4.34	2.00%	25.95%
ProtoNet Transformer	26.53	29.32%	24.22%

4.4 Qualitative Analysis

In order to check what the model has learned, I plotted the prototypes from some of the better performing models, as seen in Figure 4. This plot is a projection onto the plane using standard Principal Component Analysis (PCA) [17].

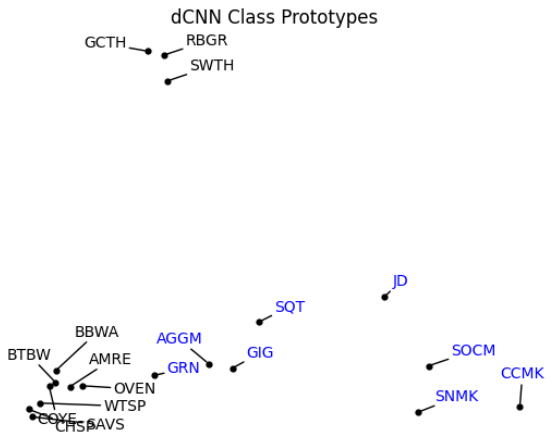


Figure 4: Dilated Convolutional Network Learned Prototypes

Bird species code abbreviations:

AMRE: American Redstart
BBWA: Bay-breasted Warbler
BTBW: Black-throated Blue Warbler
COYE: Common Yellowthroat
CHSP: Chipping Sparrow
GCTH: Gray-cheeked Thrush
OVEN: Ovenbird (Warbler)
RBGR: Rose-breasted Grosbeak
SAVS: Savannah Sparrow
SWTH: Swainson's Thrush
WTSP: White-throated Sparrow

These were trained on all of the DCASE data, so they learned prototypes for the jackdaws (JD),

heyenas (bottom right cluster), and meerkats (middle cluster). It's reassuring that the ProtoNet learns these classes! It can clearly tell each animal apart very well. In addition, we see that it's separating out GCTH, RBGR, and SWTH, which all have very similar calls. All 3 have longer calls that start out with high chirps, followed by a slightly lower pitch, and again at the higher pitch. In addition, the pitches are very, very similar, even though "prosody" is not quite the same.

It is a bit of a letdown that JD is not closer to the other birds. This is very interesting to me. I think that jackdaws have a fairly different call than the rest on this list, but not different enough to be that far away. This could be due to a difference in data collection, background noise types, or other artifacts attributed to data collection.

4.5 Reproducing Results

This work is released in a repository: https://github.com/elipugh/dcase-few-shot-bioacoustic. Code to run experiments on GPU can be found here: Colab Notebook.

References

[1] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[3] J. Salamon, J. P. Bello, A. Farnsworth, M. Robbins, S. Keen, H. Klinck, and S. Kelling, "Towards the automatic classification of avian flight calls for bioacoustic monitoring," *PLOS ONE*, vol. 11, pp. 1–26, 11 2016.

[4] J. Salamon, J. P. Bello, A. Farnsworth, and S. Kelling, "Fusing shallow and deep learning for bioacoustic bird species classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 141–145, 2017.

[5] K. Hornik, I. Feinerer, M. Kober, and C. Buchta, "Spherical k-means clustering," *Journal of Statistical Software, Articles*, vol. 50, no. 10, pp. 1–22, 2012.

[6] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," in *Proc. IEEE ICASSP 2017*, (New Orleans, LA), 2017.

[7] V. Lostanlen, J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling, and J. P. Bello, "Per-channel energy normalization: Why and how," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39–43, 2019.

[8] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, and J. Bello, "Robust sound event detection in bioacoustic sensor networks," *PLOS ONE*, vol. 14, p. e0214168, 10 2019.

[9] M. Delcroix, K. Kinoshita, C. Yu, A. Ogawa, T. Yoshioka, and T. Nakatani, "Context adaptive deep neural networks for fast acoustic model adaptation in noisy conditions," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5270–5274, 2016.

[10] J. Cramer, V. Lostanlen, A. Farnsworth, J. Salamon, and J. P. Bello, "Chirping up the right tree: Incorporating biological taxonomies into deep bioacoustic classifiers," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 901–905, 2020.

[11] J. Pons, J. Serrà, and X. Serra, "Training neural audio classifiers with few data," *CoRR*, vol. abs/1810.10274, 2018.

[12] A. Parnami and M. Lee, "Few-shot keyword spotting with prototypical networks," 2020.

[13] S. Choi, S. Seo, B. Shin, H. Byun, M. Kersner, B. Kim, D. Kim, and S. Ha, "Temporal convolution for real-time keyword spotting on mobile devices," *CoRR*, vol. abs/1904.03814, 2019.

[14] J. Lewis, "Fast normalized cross-correlation," *Ind. Light Magic*, vol. 10, 10 2001.

[15] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[16] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015.

[17] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.