STACKED CONVOLUTIONAL AND RECURRENT NEURAL NETWORKS FOR BIRD AUDIO DETECTION

Sharath Adavanne, Emre Cakir, Konstantinos Drossos, Tuomas Virtanen

Department of Signal Processing, Tampere University of Technology

ABSTRACT

In this paper, we focus on bird audio detection in short audio segments (namely 10 seconds) using stacked convolutional and recurrent neural networks. The evaluation data for this task was recorded in an acoustic soundscape different from the development data, thus motivating to work on methods that are generic and context independent. Data augmentation and regularization methods are proposed and evaluated in this regard. Area under curve (AUC) measure is used to compare different results. Our best achieved AUC measure on five cross-validations of the development data is 95.3% and 88.41% on the unseen evaluation data.

***Index Terms*—** Bird audio detection, convolutional recurrent neural network

1. INTRODUCTION

Bird audio detection (BAD) is defined as identifying the presence or absence of bird call/tweet in a given audio recording. This task acts as a preliminary step in the automatic monitoring of biodiversity. Post identifying the presence of bird call activity, a species based classifier can recognize the bird call more accurately. In this regard, the bird audio detection challenge [1] was organized with an objective to create robust and scalable algorithms which can work on real life bioacoustics monitoring projects without any manual intervention. The challenge provided annotated and non-annotated bird call recordings. The former were selected from a wide range of field and crowd-sourced recordings and is utilized as the training dataset. The latter are recordings from a completely different geographical location and employed as the test dataset.

In this paper, we propose to use methods from the sound event detection (SED) task and adapt it to the specific problem of detecting bird calls, approaching the BAD as a SED problem. The rest of the paper is organized as follows. Acoustic features representing the harmonic and non-harmonic content of the audio used in our BAD system are discussed in Section 2. The state of the art network for SED task and its configuration for the BAD is explained and presented in Section 3. Different regularization and data augmentation techniques are studied for generalizing the BAD systems and the results are reported and discussed in section 4.

The research leading to these results has received funding from the European Research Council under the European Unions H2020 Framework Programme through ERC Grant Agreement 637422 EVERYSOUND. Part of the computations leading to these results were performed on a TITAN-X GPU donated by NVIDIA. The authors also wish to acknowledge CSC-IT Center for Science, Finland, for computational resources.

2. FEATURES

Bird call is a natural phenomenon like human speech and is used by them to communicate with birds of the same species. Just like human speech and singing, bird calls can have harmonic, non-harmonic, broadband and noisy structure [2].

In this paper, we experiment with two features and analyze their contributions. The overall harmonic and non-harmonic content of the audio is represented using the log mel-band energy feature (referred as mel in future). mel has also been shown to be effective in the general SED tasks [3]. We represent the harmonic content in an audio using the three dominant frequencies and their respective magnitudes (referred as $dom\text{-}freq$ in future). $dom\text{-}freq$ has been used as a perceptual feature in SED tasks [4] and has provided considerable improvement when used along with mel.

Both the features were extracted from Hamming window frames of 40 ms length with 50% overlap. The three $dom\text{-}freq$'s were extracted in the range of 500-8000 Hz. The extraction was done on thresholded parabolically-interpolated STFT [5] using the librosa implementation [6]. The log mel-band energy was calculated for 40 mel-bands in 0-8000 Hz range.

3. STACKED CONVOLUTIONAL AND RECURRENT NEURAL NETWORKS

In the case of general SED, the best results have been reported in [3] using stacked convolutional and recurrent neural networks (CRNN). Similar combined CRNN architecture have also been proposed in automatic speech recognition [7] and music classification [8]. The CRNN architecture exploits the combined modeling capacities of a convolutional neural network (CNN), recurrent neural network (RNN), and fully connected (FC) layer. In [4], these CRNN's were extended to accommodate multiple feature classes and the feature maps

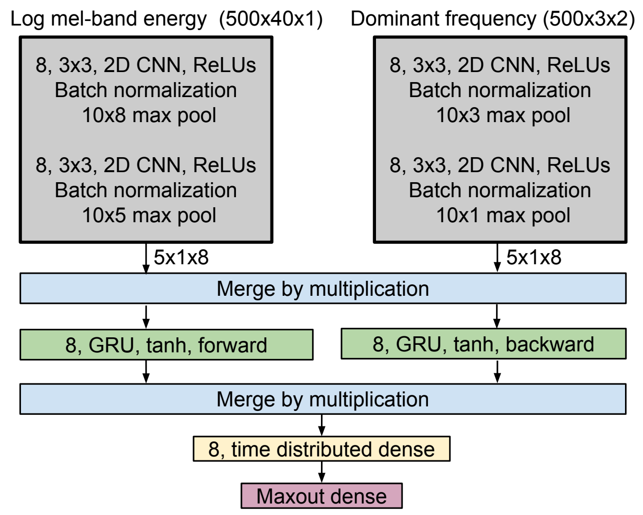

Fig. 1. Stacked convolutional and bi-directional recurrent neural networks (CBRNN) architecture for bird audio detection using multiple feature classes

from CNN's were processed using bidirectional RNN. This system was called the convolutional bidirectional recurrent neural network (CBRNN). We use this CBRNN for recognizing the presence of bird call in the audio.

Each of the feature classes, mel and $dom\text{-}freq$, is handled separately in CBRNN. T frames of 40 mel from mono channel audio are stacked into a volume of $T \times 40 \times 1$. While the three frequencies and their amplitudes of $dom\text{-}freq$ are layered into a volume of $T \times 3 \times 2$. Separate CNN's are employed to learn local shift-invariant features from each of these volumes as shown in Figure 1. A max pooling operation is performed after every CNN layer in both the axes reducing the final dimension of both the feature classes to $5 \times 1 \times N$, where N is the number of filters in the last CNN layer. We use a receptive field of 3×3 for all CNN's. The feature maps from CNN are merged using a multiplication operation and fed to bi-directional gated recurrent unit (GRU) layers followed by fully-connected time distributed dense layers. The output layer consists of a maxout dense layer with sigmoid activation function.

Batch normalization [9] was employed for all the CNN layers. The CBRNN was trained with back-propagation through time [10] using adam optimizer [11] and mean square error objective. In order to reduce overfitting of the model, early stopping was used to stop training if the area under curve (AUC) measure (Section 4.2) did not improve for 50 epochs. Dropout [12] was employed as regularizer to make the model generic and avoid overfitting to the training data. The input to the network was the complete 10-second audio, amounting to $T = 500$ frames, while the output of the network is the posterior probability in [0, 1] range, where a posterior closer to one represents the presence of bird call. Keras [13] was used to implement the neural network architecture.

Dataset	Bird call	
	present	absent
freefield1010	5755	1935
warblr	1955	6045
Total	7710	7980

Table 1. Bird audio detection challenge [1] development set statistics

4. EVALUATION AND RESULTS

4.1. Datasets

The bird audio detection challenge [1] consisted of a development and an evaluation set. These data came from three separate datasets: i) field (freefield1010), ii) crowdsourced (warblr), and iii) remote monitored (chernobyl). While the development set comprised of freefield1010 and warblr only, the evaluation set comprised of data unseen in development, predominantly coming from the chernobyl dataset. Recordings in both the sets were 10 seconds long, mono channel and sampled at 44.1 kHz. The labels for the development set were binary - bird calls present or absent. The statistics of the development set are presented in Table 1. The evaluation set consisted of 8620 audio recordings.

From the development set, we generated five cross-validations (CV) splits of 60% training, 20% validation, and 20% testing. Each split had an equal distribution of birds call present and absent. All development set results in future are the average performance on this five CV split.

For the challenge submission, the CBRNN is trained on three CV splits of 80% training and 20% validation done on development set, with equal distribution of classes in each split. For each of the CV split the trained CBRNN is evaluated on the unseen test set, and the average posterior score is submitted as the final result.

4.2. Metrics

The BAD system output is evaluated from the receiver operating characteristic (ROC) using the area under curve (AUC) measurement [14].

4.3. Results

For the hyper-parameter estimation of CBRNN, we experimented with 1 to 4 layers each of CNN, RNN and FC. The number of units for each of these layers were varied in the set of $[4, 8, 16, 32, 64, 128]$. The same dropout rate was used for all layers and varied in the set of $[0.25, 0.50, 0.75]$. The parameters were decided using a five cross-validation over the development set. The best configuration with least number of weights had two layers of CNN's with 8 filters each, one layer each of RNN with 8 units and an FC with 8 units. This configuration had only 2,600 weights and gave an AUC of over 95% on the development set. Other higher configuration of CBRNN, having up to 500,000 weights did not show any significant improvement, and the AUC score was comparable to the 2,600 weights configuration.

Similar hyperparameter experiments were done for the mel and $dom\text{-}freq$ features individually, and the same CBRNN configuration was seen to be one of the top performers on the development set with over 95% AUC for mel and around 87% for $dom\text{-}freq$. This can be accounted to the fact that mel can represent both harmonic and non-harmonic structure of a bird call. Whereas, $dom\text{-}freq$ in itself cannot justify the non-harmonic structure.

The best CBRNN configuration was seen to generalize well with a dropout of 0.75 and was seen to overfit for 0.25 and 0.50. The overfitting was observed from the training and validation AUC score plot with respect to training epochs. On employing early stopping, we control this overfitting at different drop out rates and achieve a comparable AUC of over 95% on the development set.

The CBRNN network and training configurations derived from the development set were now used for the challenge submission, for which the CBRNN's were trained on the three CV on the development set as described in section 4.1. The average of validation score of the three CV and the corresponding evaluation score for different dropout rates is presented in Table 2. We see that across the feature classes, and the drop out rates, the validation scores are comparable. While, the test scores are seen to vary about 4% across the features.

To test the statistical significance of this 4% we went through the results of the validation data. We thresholded the posterior probability of final maxout dense layer using a value of 0.5, ie, all recordings which had greater than 0.5 were flagged to have bird call, or otherwise absent. Among the 3138 validation recordings, there were 377 recordings classified wrongly. 242 of these were false positives according to the ground truth, ie, the recording was flagged to have a bird call, when it was absent. On listening through these false positives randomly, we found 37 of the 70 recordings having noticeable bird audio activity. Similarly, in false negatives, we found 7 of 30 recordings to have no bird activity. In total 42 of 100 recordings tested had wrong labels. From these, some of the recordings needed sharp ears to identify the presence and absence of birds call. A small rate of such errors are obvious in any kind of manual annotation, and the classification system has to be robust to these. Whereas, when it comes to comparing the performance of two algorithms on the same dataset, we can only claim that an algorithm is superior if the difference in performance is statistically significant. And in this scenario, it seems that 4% of the difference may not be significant enough to decide the best configuration.

Data augmentation method such as blocks mixing [15] and test mixing approaches gave marginal improvement. Individually, the improvement in AUC was consistently better with test mixing than block mixing. In the BAD challenge, the evaluation data is unseen and the recordings come from an entirely different location. This means the acoustic soundscape is different and, hence, the performance of classifier may be poor on the evaluation data. In order to teach the classifier what it can expect, we expose it to the test data by mixing it with training data. Unfortunately, we do not have the labels of test data, so we cannot mix every training recording with a random test recording. Thus, we perform the mixing only on training recordings where bird call is present (positive label). This way no matter what the test recording has, the training label will remain positive after mixing. Ideally, we can mix every training recording with each of the test recordings, but we limit ourselves to mixing once, and thereby doubling the training data for the positive class. In future, a similar augmentation method will have to be devised and performed on the negative cases, so that the classifier is equally exposed to test data ambiance for both the classes.

Features	Dropout	Validation score	Test score
mel + $dom\text{-}freq$	0.25	95.11	85.42
	0.50	94.72	86.14
	0.75	94.81	84.15
mel	0.25	95.17	88.20
	0.50	95.28	85.31
	0.75	95.30	87.35

Table 2. AUC scores on evaluation dataset

According to the AUC measures on the test data, the proposed architecture scored a best of 88.41%. This was achieved by using only the mel feature, with 0.25 dropout rate, and test mixing data augmentation.

5. CONCLUSION

A stacked convolutional and bidirectional recurrent neural network architecture (CBRNN) was proposed for bird audio detection task. Two features representative of the harmonic and non-harmonic structures of bird call in an audio were used to train the network. The evaluation data for the challenge was from a different acoustic soundscape from that of the training data. Different generalization techniques using data augmentation and regularization methods were proposed to tackle this. A challenging area under curve measure of 88.41% was achieved on this unseen evaluation data, and 95.3% on the development data.

6. REFERENCES

[1] "Bird audio detection challenge," 2016. [Online]. Available: http://machine-listening.eecs.qmul.ac.uk/bird-

audio-detection-challenge/

[2] S. Nowicki, “Bird acoustics,” in *Encyclopedia of Acoustics*, 1997.

[3] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, “Convolutional recurrent neural networks for polyphonic sound event detection,” in *IEEE/ACM TASLP Special Issue on Sound Scene and Event Analysis*, 2017, accepted for publication.

[4] S. Adavanne, P. Pertilä, and T. Virtanen, “Sound event detection using spatial features and convolutional recurrent neural network,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[5] J. O. Smith, *Sinusoidal Peak Interpolation, in Spectral Audio Signal Processing*, accessed 16.01.2017, online book, 2011 edition. [Online]. Available: https://ccrma.stanford.edu/~jos/sasp/Sinusoidal_Peak_Interpolation.htm

[6] B. McFee, M. McVicar, C. Raffel, D. Liang, O. Nieto, E. Battenberg, J. Moore, D. Ellis, R. Yamamoto, R. Bittner, D. Repetto, P. Viktorin, J. F. Santos, and A. Holovaty, “librosa: 0.4.1,” accessed 16.01.2017. [Online]. Available: http://dx.doi.org/10.5281/zenodo.32193

[7] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, long short-term memory, fully connected deep neural networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[8] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Convolutional recurrent neural networks for music classification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017,.

[9] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *CoRR*, vol. abs/1502.03167, 2015.

[10] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” in *Proceedings of the IEEE*, 1990.

[11] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *arXiv:1412.6980 [cs.LG]*, 2014.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” in *Journal of Machine Learning Research (JMLR)*, 2014.

[13] F. Chollet, “Keras,” https://github.com/fchollet/keras, 2015.

[14] “Area under curve.” [Online]. Available: https://en.wikipedia.org/wiki/Receiver_operating_characteristic#Area_under_curve

[15] G. Parascandolo, H. Huttunen, and T. Virtanen, “Recurrent neural networks for polyphonic sound event detection in real life recordings,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.