Audio Processing and Indexing
Bird Audio Detection Challenge

Andreas Papagiannis

January 2019

1 Introduction

Classifying sounds has been long used. However, since sounds can greatly vary between each other, different classifiers have been used in different sound signal, all yielding different results. The goal of this project is to build a classifier able to detect bird presence in audio files. The challenge was first posted in the Machine Listening Lab of Queen Mary University of London [3].

2 Dataset

The dataset in use is called Crowdsourced dataset (Warblr) and it comes from a UK bird-sound crowdsourcing research spinout called Warblr. It contains 8,000 ten-second smartphone audio recordings from around the UK resulting to 44 hours of audio files. The audio covers a wide distribution of UK locations and environments, and includes weather noise, traffic noise, human speech and even human bird imitations. Of course, some of the audio files contain birds chirping or singing, while others contain no birds whatsoever.

3 Pre-processing

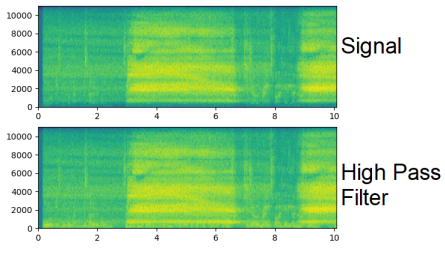

Figure 1: Top: Spectogram of a sound containing a bird, Bottom: Spectogram of the high passed version of the same sound

The frequency range of birds singing is around 3kHz to 8kHz. Since this is the information that interest as most when trying to classify between bird sounds and no bird sounds, before having their features extracted, a high pass audio filter with a cutoff frequency of 2kHz was applied to all .wav files.

In Figure 1 we can see that while the two spectograms are nearly identical (which is to be expected as they are spectograms of the same initial sound), the thick blue horizontal bottom line in the top spectogram disappears in the spectogram of the high passed sound. That is happening because the bottom spectogram represents the high passed version of the signal represented in the top spectogram. In all spectograms the y axis usually represent the frequency spectrum of the signal from low frequencies (bottom part of the y axis) to high frequencies. So the same difference in spectograms is expected to all high passed signals.

After high passing all signals, the following features were extracted from them:

- **Mel-frequency cepstral coefficients** (MFCCs), which collectively make up a Mel-frequency cepstral (MFC) and are related to the frequency content of the signal.
- **Chromagram**, this feature is also related to the frequency of the signal, It can be viewed as a scaled-to-the-frequencies-of-notes spectogram of a signal
- **Mel spectogram**, time-frequency depiction of the signal on a Mel frequency scale.
- **Spectral contrast**, another frequency based feature first applied in 2002 [2]
- **Tonal centroid features**, frequency based feature emphasizing on mid range frequencies.

These features, along with the labels containing the ground truth, were then fed to the classifier.

4 Classifier

Before fed into the classifier constructed, the prepossessed dataset was split into 30% and 70% representing the test and train set respectively.

The classifier itself is based on a neural network with three activation layers. The input layer (Softmax function), a hidden layer (Rectified Linear Unit) and an output layer (Softmax function again). As an optimizer, the Adadelta optimizer was used. Also, a dropout rate of 20% between each layers was applied. Last, categorical cross-entropy was used as loss function of the net. The classifier was then trained over 100 epochs with all aforementioned features inputted. Batch size was set to 64.

4.1 Results

Having prepossessed the data with a high pass filter and using the classifier above, training accuracy equal to 0.85 was reached along with validation accuracy equal to 0.82. Last, training loss was equal to 0.34 and validation loss equal to 0.43.

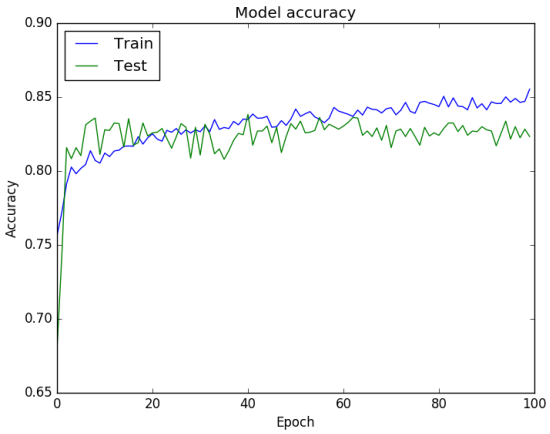

Figure 2: Model accuracy on both train and test data

Figures 2 and 3 show the classifiers accuracy as well as loss for both the training and testing datasets for all epochs the model was trained.

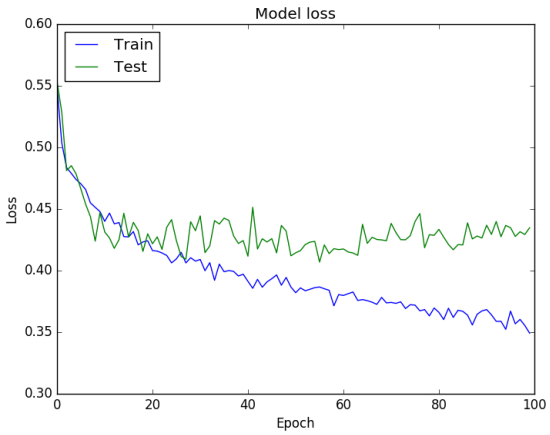

Figure 3: Model loss on both train and test data

Supplementary material, data and code:
https://github.com/dru93/birdClassification.

References - Web Resources

[1] S. Stober, T. Thomas Prätzlich, M. Müller; *Brain Beats: Tempo Extraction From EEG Data*; Conference: Proceedings of the International Conference on Music Information Retrieval (ISMIR) Jan. 2016.
http://bib.sebastianstober.de/ismir2016.pdf

[2] D. Jiang, L. Lu, H. Zhang, J. Tao, L. Cai; *Music type classification by spectral contrast feature*; Conference: Proceedings. IEEE International Conference on Multimedia and Expo Aug. 2002
http://bib.sebastianstober.de/ismir2016.pdf

[3] Bird Audio Detection challenge;
http://machine-listening.eecs.qmul.ac.uk/bird-audio-detection-challenge/

[4] Keras python package; https://keras.io/