CIAIC-BAD SYSTEM FOR DCASE2018 CHALLENGE TASK3

Technical Report

Jisheng Bai[1]*,Ru Wu*[1]*,Mou Wang*[2]*,Dexin Li*[2]*, Di Li*[2]*, Xueyu Han*[2]*,*
Qian Wang[2]*, Qing Liu*[2]*, Bolun Wang*[3]*, Zhonghua Fu*[3]*,*

[1] Northwestern Polytechnical University, Xi'an, China, {baijs}@mail.nwpu.edu.cn
[2] Northwestern Polytechnical University, Center of Intelligent Acoustics and Immersive Communications, Xi'an, China, {dexinli}@mail.nwpu.edu.cn
[3] Northwestern Polytechnical University, Audio Speech Language Processing Group, Xi'an, China,{berlloon}@gmail.com

ABSTRACT

In this technical report, we present our system for the task 3 of Detection and Classification of Acoustic Scenes and Events 2018 (DCASE2018) challenge, i.e. bird audio detection(BAD). First, log mel-spectrogram and mel-frequency cepstral coefficients (MFCC) are extracted as features. In order to improve the quality of original audio, same denoising methods are adopted, for example, adaptive denoising in Adobe Audition. Then, convolutional recurrent neural networks (CRNN) with customized activation function is used for detection. Finally, we use aforementioned features as inputs to train our CRNN model and make a fusion on three subsystems to further improve the performance. We evaluate the proposed systems on the dataset with area under the ROC curve (AUC) measure, and our best AUC score on leaderboard dataset is 85.67.

***Index Terms*—** DCASE, bird audio detection, convolutional neural network, adaptive denoising.

1. INTRODUCTION

Detecting bird sounds in audio is an important task for automatic wildlife monitoring, as well as in citizen science and audio library management [1]. Bird sound detection is a very common required first step before further analysis (e.g. classification, counting), and makes it possible to conduct work with large datasets (e.g. continuous 24h monitoring) by filtering data down to regions of interest. The task is to design a system that can return a binary decision for the presence/absence of bird sound (bird sound of any kind) given a short audio recording. The output can be just "0" or "1", but we encourage weighted/probability outputs in the continuous range [0,1] for the purposes of evaluation. For the main assessment, we will use the well-known AUC measure of classification performance.

In previous studies, convolutional neural networks (CNN) and recurrent neural network (RNN) as classifiers have recently shown improved performaces over established methods in various sound recognition tasks [2]. And CRNN has provided state-of-the-art results on various polyphonic sound event detection and audio tagging tasks [3].

In our proposed system, we firstly used spectral subtraction and adaptive denoising in Adobe Audition CS6 to denoise the original audio segments. Then, several pre-processing techniques are also tested, such as global mean removal, mean and variance standardization. Whereafter, we extract log mel-spectrogram and MFCC from the denoised audio. Finally, we make a fusion on probability.

The rest of the paper is organized as follows. In section 2, the dataset, denoising and features used in proposed system is described. In section3, we interpret the CRNN and the corresponding configuration. Experiment result is presented in Section 4. Section 5 concludes our work.

2. DATASET DENOISING AND FEATURES

2.1. Dataset

The development dataset for task 3 is divided into 3 parts, "freefield1010", "warblrb10k" and "BirdVox-DCASE-20k". For "freefield1010", a collection of 7,690 excerpts from field recordings around the world; "warblrb10k", 8,000 smartphone audio recordings from around the UK; and "BirdVox-DCASE-20k", 20,000 audio clips collected from remote monitoring units. As for evaluation datasets, it contains "warblrb10k" (a held-out set of 2,000 recordings from the same conditions as the Warblr development dataset), "Chernobyl" (6,620 audio clips collected from unattended remote monitoring equipment in the Chernobyl Exclusion Zone) and "PolandNFC" (4,000 recordings from Hanna Pamula's PhD project of monitoring autumn nocturnal bird migration). For these dataset, the common point is that most of the bird audio are mixed with noise, so denoising is important. All of the datasets contain 10-second-long WAV files (44.1 kHz mono PCM), and are manually labelled with a 0 or 1 to indicate the absence/presence of any birds within that 10-second audio clip.

2.2. Denoising

Because the bird audio cover or mix with different noise, denoising is necessary. But there are so many noise types such as rain, wind, thunder and so on, and we can't denoise with only one type of noise. So we firstly use spectral subtraction as follows:

$$D(w) = P_s(w) - \alpha P_n(w) \quad (1)$$

$$P_s^{'}(w) = \begin{cases} D(w), & if \quad D(w) > \beta P_n(w) \\ \beta P_n(w), & if \quad otherwise \end{cases} \quad (2)$$

where $P_s(w)$ is the input of the noisy audio spectrum, and $P_n(w)$ is the spectrum of the estimated noise. Those two are subtracted

so we can get the difference spectrum $D(w)$ [4]. α is subtraction factor and β is lower threshold parameter of spectrum. We use the first 0.6 second of the input audio to be the silence time, which can be called the bottom noise, and the noise is taken as an estimated noise.

Then we use Adobe Audition CS6 to adaptively denoise the audio, which is also based on spectral subtraction. However, we find it performs better than spectral subtraction.

2.3. Features

By observing the bird audio we found that most of the birds frequencies range from around 500 Hz to 9000 Hz, and a huge number of birds frequencies are between 1000 Hz to 8000 Hz. Therefore we set the mel filter parameters (fmin and fmax) according to this characteristic. Three different features are used in our system as follows:

(1) Log-mel energy: we use the denoised audio to calculate STFT with a hanning window of 23 ms, and the hop length is set of 11.5 ms. Then 80 log mel-band energy features from 250 Hz to 8 kHz and 862 frames are extracted from each file. So each 10s file are transformed into a 80*862 matrix.

(2) Log-mel spectrogram: we plot the log mel matrixs and save them as pictures. And the pixels for the pictures are 335*217. For training,we read these pictures as inputs.

(3) MFCC: we use 117 dimensions including first and second order difference. A hanning window is 23 ms and the hop length is 10 ms. 68-band mel filter banks are from 1000 Hz to 8700 Hz.

3. CRNN IN BAD SYSTEM

3.1. Models

CRNN was used in our BAD system. Our model details are presented as four parts:

(1) Features are fed into 4 convolutional layers with 32, 64, 96, 128 feature maps, filter kernels are 3*3 and activations are changed learnable gated activation function. Non-overlapping pooling was used to pool frequency axis into 1 dimension. However, we didn't use pooling over time axis. Otherwise, it will loss information about time. But for MFCC features, we apply $2*2$ pooling over two axes.

(2) For log-mel features, the last layer of feature maps of CNNs is 128, and we stack feature maps over frequency axis and feed it into 2 GRUs (128 units) for log-mel feature, because frequency axis has been pooling to 1 and it means finding the most representative feature over frequency axis. And a time maxpooling layer is used after GRUs and each file is extracted to a 128*1 feature map. This can find the most representative feature over time axis. But it will loss difference information so we maintain 7 dimensions for MFCC features.

(3) We use fully-connected layers in the end.For log-mel features,there are 2 fully-connected layers,the units are $\{128,2\}$ to get probabilities between 0 to 1. But for MFCC, there are 3 fully-connected layers, the units are $\{1024,512,2\}$.

In the training, we use batch-normalization [5] and dropout [6] to regulate. But we found that batch-normalization didn't work for some situations. Tensorflow has been used to establish our model and some python libraries such as librosa [7] are also helpful.

And for these features, the detail model parameters are as follows in table 1 ('-' means no layer).

Table 1: Network architectures of three different features.

Features	Log mel energy	Log mel spectrogram	MFCC
Input	1*862*80	4*335*217	1*990*117
Conv	32*862*80	32*335*217	32*990*117
Pool	32*862*40	32*335*43	32*495*58
Conv	64*862*40	64*335*43	64*495*58
Pool	64*862*10	64*335*10	64*247*29
Conv	96*862*10	96*335*10	96*247*29
Pool	96*862*2	96*335*2	96*123*14
Conv	128*862*2	128*335*2	128*123*14
Pool	128*862*1	128*335*1	128*61*7
GRU*2	128*862*1	128*335*1	128*61*7
Time pool	128*1*1		-
Dense	128		1024
Dense	-		512
Dense	2		

3.2. Changed learnable gated activation function

Using the learnable gated activation function for training the neural network in DCASE2017 task4 is presented by Xu [8]. And the results shows this activation function can make great improvement for sound detection. So we tried his method and we proposed a variant of his learnable gated activation function.

$$\mathbf{Y} = \mathbf{X} * \mathbf{W} + b \quad (3)$$

$$\mathbf{Z1} = \mathrm{ReLU}(\mathbf{Y}) \quad (4)$$

$$\mathbf{Z2} = \mathrm{sigmoid}(\mathbf{Y}) \quad (5)$$

$$\mathbf{Z} = \mathbf{Z1} \bigotimes \mathbf{Z2} \quad (6)$$

The $\mathbf{X}$ is the input feature map, $\mathbf{W}$ and b is weights and biases in each layer. Then we applied ReLU and sigmoid activation function to get the output **Z1** and **Z2**. And we multiply **Z1** and **Z2** to get the final output $\mathbf{Z}$. This changed gated activation function is applied in convolutional layers and full-connected layers, but it cost error if it is used in GRUs. We have no idea about this problem.

3.3. Fusion

For fusion, because we have three different features and each type of feature has its own focus, we fused these results and get better results. The experiment results are shown in next section.

4. EXPERIMENTAL RESULTS

4.1. Experiment setting

We used the three different features and trained three different models. During the training, back-propagation and Adam optimizer

Table 2: The results of AUC score on development and evaluation datasets.

Dataset	Features	Score	Fusions
Development	Log-mel energy	94.6	
	Log-mel spectrogram	94.9	
	MFCC	94.5	
Evaluation	Log-mel energy	79.56	85.67
	Log-mel spectrogram	79.68	
	MFCC	80.67	

with learning rate of 0.0001 are used.

4.2. Experiment results

The score of development is 20% of the development dataset as validation dataset under AUC score which applies python sklearn library. We obtain over 94 on development dataset, but the scores of evaluation dataset(1000 of the whole 12,620 evaluation dataset) are about 80. Surprisingly, we fused these results by averaging them and we finally obtained 85.67 on evaluation dataset.

Though log mel matrices are more accuracy but results shows that saving the matrices as png pictures also performs well in BAD task.

5. CONCLUSIONS

In this technical report, we propose using CRNN for BAD task. And we present a new activation function, its a changed gated activation function of Xu. At last, we get 85.67 under AUC score on evaluation dataset.

6. REFERENCES

[1] `http://dcase.community/challenge2018/task-bird-audio-detection.`

[2] E. Cakir, S. Adavanne, G. Parascandolo, K. Drossos, T. Virtanen, "Convolutional recurrent neural networks for bird audio detection", Bird Audio Detection Challenge Tech. Rep., 2017

[3] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar,E. Battenberg, and O. Nieto, librosa: Audio and music sig-nal analysis in python,in Proceedings of the 14th Python in Science Conference,2015

[4] Berouti, M., R. Schwartz, and J. Makhoul. "Enhancement of speech corrupted by acoustic noise." Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP IEEE, 2003:208-211.

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

[6] 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.

[7] Gui.Audio feature extraction – use of librosa Toolkit.http://www.cnblogs.com/xingshansi/p/6816308.html

[8] Y. Xu, Q. Kong, W. Wang and M. D. Plumbley, Large-scale weakly supervised audio classification using gated convolutional neural network, in Proc. IEEE ICASSP, 2018.