MULTI-SCALE RESIDUAL CRNN WITH DATA AUGMENTATION FOR DCASE 2020 TASK 4

Technical Report

Maolin Tang, Longyin Guo, Yanqiu Zhang, Weiran Yan, Qijun Zhao

Sichuan University
College of Computer Science
ChengDu, SiChuan, China

ABSTRACT

In this technical report, we present our method for task 4 of DCASE 2020 challenge (Sound event detection and separation in domestic environments). The goal of the task is to evaluate systems for the detection of sound events using real data either weakly labeled or unlabeled and simulated data that is strongly labeled (with timestamps). We find that models perform well on synthetic data, but may not perform well on real data. We thus improve the baseline [1] by using a variety of data augmentation methods and synthesizing more complex synthetic data for training. Moreover, we present multi-scale residual convolutional recurrent neural network (CRNN) to solve the problem of multi-scale detection. The promising results on the validation set prove the effectiveness of our method.

Index Terms— Sound event detection, Mean teacher, Residual block, Median window, Audio synthesis, Data augmentation

1. INTRODUCTION

One of the DCASE 2020 challenges is how to better use synthetic audio for training. In [2], it is said that the model may overfit on the synthetic data, resulting in poor performance on real audio, because of the unbalance between synthetic audio and real audio in the labeled data. To solve this problem, we synthesize more complex synthetic audio and use a variety of data augmentation methods to expand the scale of the weakly labeled data set. In addition, we also use the method of spectrum augmentation to reduce overfitting. Like the baseline method in [1], we use mean teacher [3] for semi-supervised learning. In order to solve the problem of multi-scale detection, we modify the ResNet [4] module to extract convolution kernels of different sizes for features of different scales. To reduce false positives, we replace attention pooling function with linear softmax function. The evaluation results on the validation set prove the superiority of our method over the baseline method.

2. METHOD

In this section we will introduce our method, particularly its novel components with respect to the baseline method in [1], including linear softmax, multi-scale residual block, data augmentation and synthesis.

2.1. Overview

Our model follows the CRNN structure. In the CNN part, we apply several multi-scale residual blocks we presented to extract multi-scale local features. In the RNN part, we apply two Bi-GRU to extract temporal features. Afterwards, a Dense layer is applied to obtain the event frame-level probability. Finally, linear softmax is utilized to obtain event recording-level probability. At the same time, we use a variety of data augmentation methods and audio synthesis to help model training.

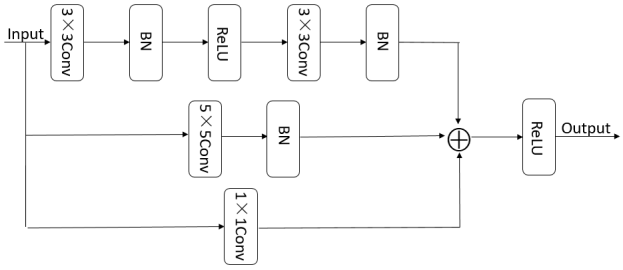

Figure 1: Illustration of Multi-scale residual block.

2.2. Multi-scale residual block

The traditional convolutional neural network(CNN) module is prone to the vanishing gradient problem if the network is too deep, which makes it difficult for the model to converge. ResNet [5] used shortcut connections to make the gradient better propagate from the back layer to the front layer during back propagation. It can accelerate the process of training. At the same time, the length of events and the number of occurrences in this task are not fixed. This will cause the model to work with inconsistent accuracy for events of different scales. To solve this problem, we add a branch in basic residual block. This branch contains convolution kernels of different sizes, so that the model can extract richer and multi-scale features. See Figure 1.

2.3. Linear softmax

Wang et al. [5] compared five different types of pooling functions in the multiple instance learning (MIL) framework for sound event detection (SED), namely Max pooling, Average pooling, Linear softmax, Exponential softmax and Attention pooling, found that Attention pooling would cause too many false positives, while Linear softmax performs the best. Linear softmax is defined as follows:

$$y = \frac{\sum_i y_i^2}{\sum_i y_i} \tag{1}$$

where y_i is the predicted probability of the i^{th} frame of an event, and y is the aggregated recording-level probability of the same event.

2.4. Data augmentation

We use three traditional data augmentation method to augment weakly labeled data: TimeStretch, PitchShift, TimeShift which all directly operate on the original audio data. TimeStretch is to stretch or shorten the duration of the audio in the time series while keeping the original audio shape basically unchanged (that is, when the pitch does not change). The principle is to use a phase vocoder, after a short-time Fourier transform, to speed up or slow down the rate by a factor, and then transform it back to the time domain. PitchShift is to increase or decrease the pitch of the original audio in semitone units while keeping the original audio time series unchanged. TimeShift refers to the shift in the time series, which is divided into two types, placing the former part of the time series at the end and placing the latter part of the time series at the front.

SpecAugment [6] includes three data augmentation methods, warping the features, masking blocks of frequency channels, and masking blocks of time steps. In our method, we use frequency masks for data augmentation. All augmentations of SpecAugment are directly operated on the audio spectrogram, which can save a lot of calculation time. The frequency mask adds some masks to the frequency channels of the spectrogram to augment the data. Frequency masking is applied so that f consecutive mel frequency channels $[f_0, f_0 + f)$ are masked, where f is first chosen from a uniform distribution from 0 to the frequency mask parameter F, and f_0 is chosen from $[0, v - f)$. v is the number of mel frequency channels. Usually the mask is to make this part of the frequency channels mean value or directly set to zero, and the mean value of the entire log mel spectrum is taken in our experiment. Unlike the traditional audio data augmentation methods, which directly operate on the original audio, this frequency mask data augmentation method only operates on a specific part, ensuring that the complete information of the remaining parts is retained without distortion. The audio augmentation in this way retains the information of the original audio as much as possible, but it is different from the original audio and realizes the augment operation.

2.5. Audio synthesis

It is well known that a model that performs well on synthesized audio may not perform well on real audio. This is because the model overfits on synthesized audio of low complexity. Therefore, we synthesize more complex synthesized audio to reduce overfitting. The synthesis uses the Scaper [7], a public toolkit for audio synthesis and augmentation. See specific parameters in Section 3.1.

3. EVALUATION RESULTS

3.1. Dataset

Among the real audio of the development dataset in this challenge, the training set contains $1,578$ weakly labeled audio, $14,412$ unlabeled audio, and the validation set contains $1,168$ audio. Each audio lasts for $10s$, and there are 10 types of sounds, including Speech, Dog, Cat, Alarm/bell/ringing, Dishes, Frying, Blender, Running water, Vacuum cleaner, and Electric shaver/toothbrush.

Table 1: F1-score for the baseline and presented methods.

	Event-based Macro F-score (%)	PSDS macro F-score (%)
Baseline	34.80	60.00
Model1	46.62	66.58
Model2	48.43	68.55
Model3	47.80	69.93
Model4	48.97	68.01

There are $2,060$ background files from SINS and $1,009$ foreground from Freesound for synthesizing audio. The reference decibel is set to -35 dB, the polyphony maximum is limited to 3, the FBSNR range is set to 2-30 dB, and a total of $3,237$ audio segments are synthesized.

3.2. Experimental setup

The audio is resampled to $22,050$ Hz and Log-Mel spectrogram is extracted from audio clips by 128-bin, $2,048$-window, and 255-hop. The spectrogram is used as the input of the system with a size of 864×128. The CNN part contains 7 multi-scale residual blocks, for which the number of filters and pooling size are respectively [16, 32, 64, 128, 128, 128, 128] and [[2, 2], [2, 2], [2, 2], [1, 2], [1, 2], [1, 2], [1, 2]]. The RNN part is the same as the baseline.

Like baseline, median filters for different events are also used in post-processing with different window sizes: Alarm/bell/ringing, Dishes, Dog and Speech is set to 2 ($0.2s$), Blender and Cat is set to 6 ($0.6s$), Electric shaver/toothbrush, Frying and Vacuum cleaner is set to 42 ($3.9s$) and Running water is set to 16 ($1.5s$).

The compression range of TimeStretch is set to [0.8, 1.5]. The pitch change range of PitchShift is set to [-2, 2] semitones and time shift range is set to [-0.2, 0.2]. The frequency mask uses four random masks of size in [0, 10].

3.3. Result

We submit a total of 4 models. Model 1 uses Linear softmax, basic residual block and Frequency mask. Model 2 adds the augmentation of weakly labeled data on the basis of Model 1. Model 3 replaces the basic residual block in the first model with the Multi-scale residual block mentioned in this report, and Model 4 adds the augmentation of weakly labeled data to Model 3. The results of these models on the validation set are shown in the Table 1. It can be seen from the results that our proposed methods are much higher than the baseline. What's more, Augmenting the weakly labeled data or replacing the basic residual block with multi-scale residual block can both improve on Event-based Macro F-score. Model 3 performs best on PSDS macro F-score.

4. CONCLUSION

In this report, we presented an improved residual CRNN for multi-scale sound event detection. We used a variety of data augmentation methods to solve unbalanced dataset problem and synthesized more complex data. Finally, Evaluation results show that our method can obtain obviously higher accuracy than the baseline.

5. REFERENCES

[1] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for dcase 2019 task 4," Orange Labs Lannion, France, Tech. Rep., June 2019.

[2] L. Lin and X. Wang, "Guided learning convolution system for dcase 2019 task 4," Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, Tech. Rep., June 2019.

[3] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, 2017, pp. 1195–1204.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 31–35.

[6] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[7] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2017, pp. 344–348.