COMBINED SOUND EVENT DETECTION AND SOUND EVENT SEPARATION NETWORKS FOR DCASE 2021 TASK 4

Technical Report

Gang Liu,Zhuang Zhuang Liu,Jun Yan Fang,Yi Liu,Ming Kun Zhou
Beijing University of Posts and Telecommunications
Beijing,China
{liugang, liuzhuangzhuang2345, fangjunyan,ly32203918,mikan_zhou}@bupt.edu.cn

ABSTRACT

Audio tagging aims to assign one or more labels to the audio clip. In this paper, we proposed our solutions applied to our submission for DCASE2021 Task4. The target of the systems is to provide not only the event class but also the event time localization given that multiple events can be present in an audio recording [1]. We present a convolutional recurrent neural network (CRNN) with two recurrent neural network (RNN) classifiers sharing the same preprocessing convolutional neural network (CNN). Both recurrent networks perform audio tagging. One is processing the input audio signal in forward direction and the other in backward direction. We also use a spatial attention layer which called Fcanet to improve our system. We also make an independent system to achieve sound event speration.
Index Terms— CRNN, Fcanet, Sound event separation

1. INTRODUCTION

Sound event detection (SED) is recently an active research topic in the areas of signal processing for machine learning. DCASE task 4 [1] aims to classify not only the sound event classes but also the event time boundaries. The baseline SED approach inspired by the mean-teacher model [2] relies on convolutional-recurrent neural network (CRNN) that has shown good capability of identifying the sound events with weakly labelled and unlabeled training data. In order to explore the possibility of improvement due to source separation, the participants are encouraged to develop a SED system combined with a sound event separation (SES) network.

2. PROPOSED FRAMEWORK

In the sections to follow, we describe the audio features, loss functions, network architectures and attention layer. We also will introduce how we deal with the labels offered by the official.

2.1. Input

Recordings are resampled to 16000 Hz and to generate mel spectrogram with a Hanning window size of 1024 and hop length of 323 samples. Mel filters which band is 1024 are used to transformed STFT spectrogram to mel spectrogram, and frequencies lower than 0 Hz and beyond 8000 Hz are removed.

2.2. Network Architecture

We use CRNN as our baseline. And the Description of convolutional neural network architecture has been list in the table

Layers
input
3×3 Conv(stride-1, pad-1)-16-BN-RELU
2×2 AvgPool(stride-2)
3×3 Conv(stride-1, pad-1)-32-BN-RELU
2×2 AvgPool(stride-2)
3×3 Conv(stride-1, pad-1)-64-BN-RELU
1×2 AvgPool(stride-2)
3×3 Conv(stride-1, pad-1)-128-BN-RELU
1×2 AvgPool(stride-2)
3×3 Conv(stride-1, pad-1)-128-BN-RELU
1×2 AvgPool(stride-2)
3×3 Conv(stride-1, pad-1)-128-BN-RELU
1×2 AvgPool(stride-2)
3×3 Conv(stride-1, pad-1)-128-BN-RELU
1×2 AvgPool(stride-2)
Attention Layer
RNN
FC

Table1: Description of convolutional neural network architecture

2.3. Loss Function

We regarded the multi-class and multi-label task as many binary problems. So we used binary cross loss function, which is offered by torch.

$$\text{loss} = -\sum \hat{y_i} log y_i + (1 - \hat{y_i}) \log(1 - y_i)$$

2.4. Attention Layer

Channel attention has been proved to be effective to improve the system. In our system, we used a new approach instead of global average pooling to calculate each channel's weight. It has been improved that the global average

pooling is a special situation of DCT and it will loss much information[3]. We add more frequency information in our attention layer. And in our experiment we find it will be best when we combine all low frequency in the attention layer.

3. SOUND SEPARATION

In the actual situation, audio events will inevitably overlap, and the performance of the method only detected will drastically decrease. In order to solve this problem, we propose to add Sound Separation module to improve the performance of audio event detection, which is also corresponding to DCASE2021 Task4.

3.1. Input

Time domain audio input with sampling rate of 16 kHz. In order to provide available data to the separation system, we get the unlabeled clean source and background sound from the FUSS dataset, and get their label from the CSV file of FSD50K. We use the same strategy as [4]. The Scaper is used to get time-aligned mixed audio. Each mixed audio is composed of 2-4 target sound sources and background sound, and its duration is set to 10s. As different audio categories have different durations, the audio longer than 10s will be randomly intercepted and mixed with 10s fragments. For the audio less than 10s long, let them uniform distribute in the mixed audio.

3.2. Model

Wave-U-net [5] is a time-domain implementation of U-net in music source separation, and can be used to model the dependence of mask and signal structural characteristics at different scales.

A diagram of the Wave-U-Net architecture is shown in Figure 1. It computes an increasing number of higher-level features on coarser time scales using down-sampling (DS) blocks. These features are combined with the earlier computed local, high-resolution features using up-sampling (US) blocks, yielding multi-scale features which are used for making predictions. The network has L levels in total, with each successive level operating at half the time resolution as the previous one. For K sources to be estimated, the model returns predictions in the interval (−1,1), one for each source audio sample. The detailed architecture is shown in Table 2.

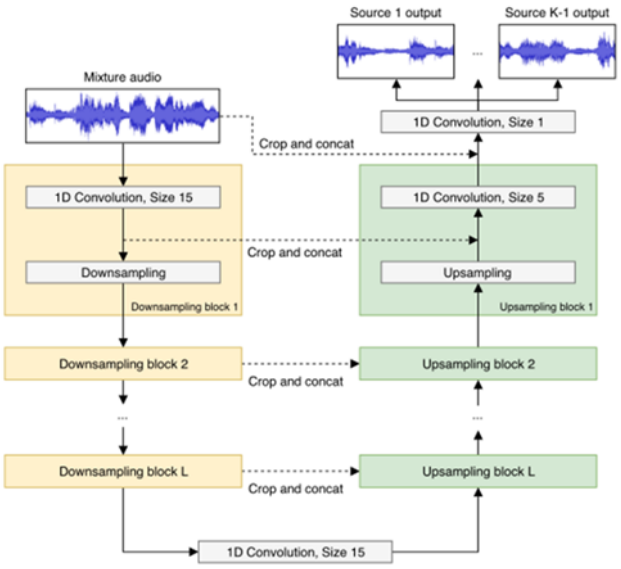

Fig 1. Architecture of wave-u-net.

Block	Operation	Shape
	Input	$(16384, 1)$
DS, repeated for $i = 1, \ldots, L$	$\texttt{Conv1D}(F_c \cdot i, f_d)$ $\texttt{Decimate}$	$(4, 288)$
	$\texttt{Conv1D}(F_c \cdot (L+1), f_d)$	$(4, 312)$
US, repeated for $i = L, \ldots, 1$	$\texttt{Upsample}$ $\texttt{Concat}$(DS block i) $\texttt{Conv1D}(F_c \cdot i, f_u)$	$(16834, 24)$
	$\texttt{Concat}$(Input) $\texttt{Conv1D}(K, 1)$	$(16834, 25)$ $(16834, 2)$

Table 2. Block diagram of the base architecture.

3.3. Loss function

We designed Three kinds of loss functions:

(1) The sum of MSE deviation of each separated signal and its corresponding actual value is calculated simply from the recovery of each sound source signal;

$$L_1 = \sum_{i=1}^{10} MSE(y_i, s_i)$$

(2) As the signal distribution is sparse, it is necessary to weaken the influence of the mute term and give a small weight (α=0.001);

$$L_2 = \sum MSE(y_{target}, s_{target}) + \alpha \sum MSE(y_{no\_tag}, s_{no\_tag})$$

(3) Previously, only single signals were considered to be consistent with the actual situation, but mixed consistency should be considered at the same time, that is, the sum of the output audio is consistent with the mixed input.

$$L_3 = \frac{1}{10} \sum_{i=1}^{10} MSE(y_i, s_i) + MSE(\sum_{i=1}^{10} y_i, x)$$

3.4. D-vector

The speaker encoder aims to generate a speaker embedding from a target speaker audio sample. This system is based on the system in [6], which is a generalized end-to-end The three-layer LSTM network trained by Loss takes log_mel spectrogram with a window length of 1600ms as the input and outputs the speaker embedding as the D-vector, which is fixed as the 256-dimensional input. In order to calculate the D-vector of each segment of discourse, 50% of the sliding Windows are extracted and used for regularization and average.
In this condition, we use the VGG network with the structure shown in Table 3 to generate D-vectors, which are embedded into the bottleneck layer of Wave-U-Net as category representation.

Layers
Log_mel input
{3*3, 64, BN, ReLU}
{3*3, 64, BN, ReLU, avg_pooling_size=(2, 2)}
{3*3, 128, BN, ReLU}
{3*3, 128, BN, ReLU, avg_pooling_size=(2, 2)}
{3*3, 256, BN, ReLU}
{3*3, 256, BN, ReLU, avg_pooling_size=(2, 2)}
{3*3, 512, BN, ReLU}
{3*3, 512, BN, ReLU, avg_pooling_size=(2, 2)}
Frequency_global_avg_pooling
Time_global_avg_pooling + Time_global_max_pooling

Table 3. Architecture of D-vector generator.

3.5. Training Strategies

Since the 2-4 target classes we separated are within the 10 known classes, the separation model output by the 10 target classes is fixed. First, a separation model is trained with the data we synthesized, and then the ten target audio sources are separated by the separation model. Secondly, the classification model is trained by DCASE2021_TASK4_DESED. Finally, the target sound source output from the separation model is used as the input of the classification model for fine-tuning to get the final model.

4. CONCLUSION

In this paper, we introduced our network used in the task. We introduce a new approach to calculate channel attention. We use DCT instead of global average pooling to obtain more information from the network. In addition, we propose a detection method benefit from sound separation model, which is more suitable for actual situation.

5. REFERENCES

[1] http://dcase.community/challenge2021/task-sound-event-detection-and-separation-in-domestic-environments

[2] L. Delphin-Poulat and C. Plapous, “Mean teacher with data augmentation for dcase 2019 task 4,” Technical Report, Orange Labs Lannion, France, June 2019.

[3] Qin, Z., Zhang, P., Wu, F., and Li, X., “FcaNet: Frequency Channel Attention Networks”, arXiv e-prints, 2020.

[4] S. Wisdom, H. Erdogan, D. P . W. Ellis, R. Serizel, N. Turpault, E. Fonseca, J. Salamon, P . Seetharaman, and J. R. Hershey, “What’s all the FUSS about free universal sound separation data ?” In preparation, 2020.

[5] Daniel Stoller, Sebastian Ewert, and Simon Dixon, “Wave-u-net: A multi-scale neural network for end-to-end audio source separation,” arXiv:1806.03185 [cs,eess, stat], 2018.

[6] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. Lopez Moreno, “ VoiceFilter: Targeted voice separation by speaker-conditioned spectrogram masking,” in Proc. Interspeech, 2019, pp. 2728–2732