Semi-Supervised Sound Event Detection System Based on Complex Convolutional Recurrent Neural Network

Technical Report

Hong Lyu, Qianhua He

School of Electronic and Information Engineering,
South China University of Technology, Guangzhou, China
lvhong_2023@163.com, eeqhhe@scut.edu.cn

ABSTRACT

This report describes the system we proposed for Task 4 of DCASE 2024. To investigate the impact of complex information on sound event detection tasks, we designed a system based on Complex Convolutional Recurrent Neural Network[1] for semi-supervised Sound Event Detection (CCRN-SED). We utilized the Mean Teacher[2] for semi-supervised learning, which can address the challenge of unlabeled data. In addition, we use BEATs pretrained model[3] to extract information from data outside the development set. The optimal PSDS1 and mean pAUC of CCRN-SED on the development test set are 0.508 and 0.693.

Index Terms— Complex Convolutional Recurrent Neural Network, Mean Teacher, Sound Event Detection

1. INTRODUCTION

The purpose of Sound Event Detection (SED) is to identify the class of vocalization events and the corresponding start and end times in the audio signal. SED can be applied in practical scenarios such as smart homes[4], traffic monitoring[5], and industrial production[6]. Currently, audio datasets with strong labels are not as rich as those in domains such as speech and image. To enhance the amount of data available for model training, models can be trained using a large number of unlabeled samples through unsupervised or semi-supervised learning approaches.

Convolutional Recurrent Neural Network (CRNN) is a general model architecture used for SED systems[7-9], where the CNN module effectively extracts local information from the feature map, and the RNN module captures temporal information in the audio, enabling the extraction of contextually relevant features. The Mean Teacher semi-supervised learning method enables the training of SED systems using weakly labeled and unlabeled data[10].

Complex Convolutional Recurrent Neural Network (CCRN) has demonstrated significant performance in Speech Enhancement[1]. This report applied CCRN to the field of SED, which gives the SED systems the ability of complex number operations to process both amplitude and phase information of audio. Besides, the Mean Teacher method and the BEATs pretrained model can significantly increase the amount of data learned by the system.

Following this introduction, Section 2 proposes the CCNN-SED method with the BEATs and Mean Teacher. Section 3 introduces the experiment on the development dataset. Section 4 concludes this report.

2. METHOD

The architecture of CCRN-SED is shown in Figure 1, which mainly contains the Complex CNN module, the Complex LSTM module and the Output module. The input is the audio time-domain waveform sampled at a frequency of 16kHz. The model produces two outputs: the strong label and the weak label. The specific structure of the three main modules and the system processing flow are described in detail below.

In Figure 1, the audio is first min-max normalized and then put into Conv STFT and BEATs. The complex spectrogram is obtained through Conv STFT, and the embedding including the information from data outside the experimental dataset is extracted through BEATs. Next, the complex spectrogram is fed into the Complex CNN module for deeper feature extraction. The BEATs embedding is then resized to match the output of the Complex CNN module in the time dimension through a pooling layer, and then it is concatenated with the complex spectrogram along the channel dimension. Third, the combined feature is passed through a FNN layer to obtain the aggregated output. Finally, the aggregated output is processed through the Complex LSTM module, the Output module generate predictions for both the strong label and weak label.

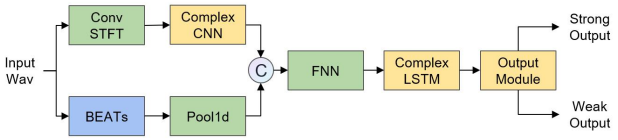

Figure 1: CCRN-SED network architecture.

2.1. Complex CNN module

Figure 2 illustrates the structure of the Complex CNN module. The Complex CNN module consists of six stacked Complex CNN blocks. Each block is composed of a sequence of ComplexConv2d, BatchNorm2d, and PReLU. Additionally, the last two blocks include an AvgPool2d layer. ComplexConv2d processes both the channel and feature dimensions, extracting information from the feature dimension and passing it to the channel dimension. AvgPool2d processes the feature and time dimen-

sions, compressing the size of the time dimension from 632 to 158. The convolution kernel size of ComplexConv is 5 × 2, with a stride of 2 × 1, and the pooling window size is 4 × 2. The number of output channels for the six blocks is [32, 64, 128, 128, 128, 128], and the size of the output feature dimension is [512, 256, 128, 64, 8, 1].

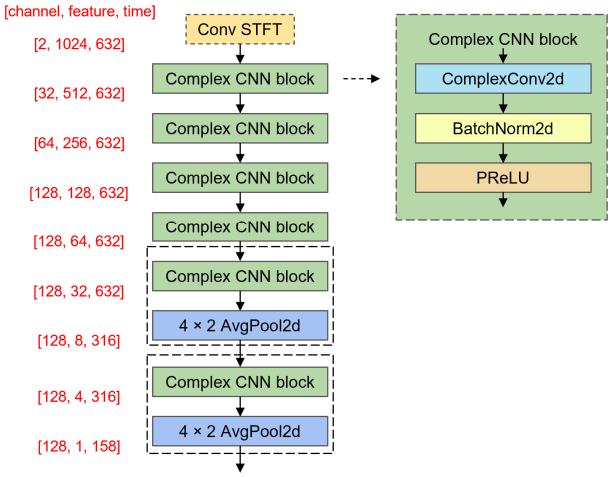

Figure 2: Structure of the Complex CNN module.

2.2. Complex LSTM module

The Complex LSTM module consists of two stacked Complex LSTM, each with a hidden state count of 384. System 1, proposed in this paper, uses a standard CLSTM, while System 2 employs a bi-directional CLSTM. The rest of the structure remains the same for both systems.

Figure 3 illustrates the structure of the output module. The output module consists of FNN and activation function. The input channel dimension size corresponds to the output dimension of the CLSTM module (i.e., the number of hidden states), while the output channel dimension size corresponds to the number of event categories (i.e., 27). The module has two output heads: a strongly labeled output and a weakly labeled output. The strongly labeled output head includes an FNN layer followed by a Sigmoid activation function, with the output time dimension size matching the number of frames. The weakly labeled output head includes an FNN layer, a Softmax activation function, and an attention layer, with the output time dimension size being one. The strong output indicates the possible sound events and corresponding start and end times for a segment of audio, while the weak output gives only the possible events for a segment of audio.

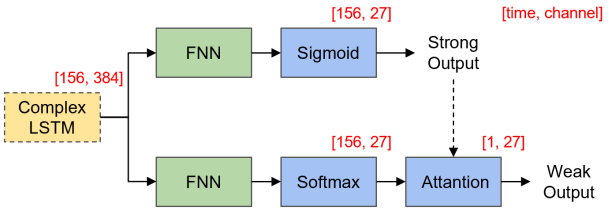

Figure 3: Structure of the output module.

2.3. System processing flow

The BEATs model was pre-trained on AudioSet-2M[3] and was not fine-tuned in the experiments reported here; it was solely used for embedding extraction. The network architecture for both the teacher model and the student model is CCRN-SED. The teacher model initially has the same parameters as the student model. Subsequently, parameters of the teacher model are exponential moving average (EMA) of the parameters of the student model, which can smooth parameter fluctuations and thus improve the stability of the model[2]. Figure 4 illustrates the overall processing flow of the system.

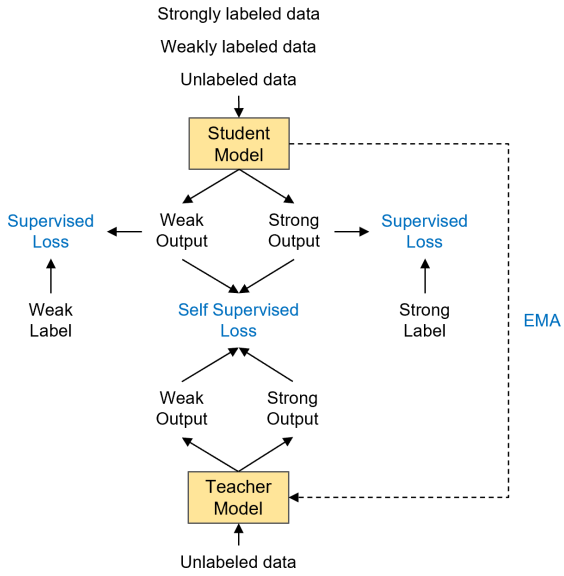

Figure 4: The overall processing flow of the system.

First, the strongly labeled data and the weakly labeled data are passed through the student model to obtain the strongly labeled and weakly labeled predictions, which are used to compute supervised loss. Next, the unlabeled data is passed through the teacher model and the student model to obtain the prediction results for both types of labels, which are used to compute self-supervised loss. Third, the parameters of the student model are updated by backpropagation. Finally, the parameters of the teacher model are updated by EMA.

3. EXPERIMENT

3.1. Experimental Dataset

The experiments were conducted with the official dataset provided by DCASE, which includes the DESED dataset[11] and the MAESTRO Real dataset[12]. DESED consists of two parts: the audio sample recorded in domestic environment and the sample synthesized by using Scaper. MAESTRO Real consists of audio samples from different realistic acoustic scenes. The dataset is divided into development set and evaluation set. The development training set includes four subsets: weakly labeled training set, unlabeled in-domain training set, synthetic strongly labeled

set, soft labeled training set. It contains a total of 35,582 mono audio instances of 10 seconds with 27 classes of sound events. The development validation set and test set contain 3,407 and 4,642 instances, respectively. The evaluation set contains 2,200 instances.

3.2. Experimental Settings

The data is downsampled to 16kHz and extracted by Conv STFT to obtain a complex spectrogram with a frame length of 128ms and a frame shift of 16ms.

We use an RTX 4090 D for our experiments. System 1 and System 2 have epochs of 200 and 220, respectively. The batch size is set to 30, and the optimizer is Adam with β_1=0.9 and β_2=0.999. The learning rate tuning strategy is Exponential Warmup, which reaches a maximum value of 0.001 when epoch is 50, then maintains that learning rate, and starts to decrease when epoch is 100. Supervised loss and self-supervised loss are BCELoss and MSELoss, respectively. Data augmentation is performed using soft mixup, and the dropout rate is set to 0.2.

3.3. Experimental Results

We use PSDS1 and mean pAUC as the evaluation metrics. The experimental results of the different systems on the development test set are shown in Table 1. Official Baseline[13] and CCRN-SED use the same BEATs pretrained model, data augmentation and semi-supervised strategy. The only difference is that Baseline uses the CRNN architecture. CCRN-SED-2 achieves the best PSDS1 score, while official Baseline achieves the highest mean pAUC. CCRN-SED-2 has better detection accuracy and robustness in general, while Baseline has better detection performance in scenarios where false alarm rate need to be controlled.

Tabel 1: Results of different systems on the development test set

System	PSDS1	mean pAUC
Baseline	0.490 ± 0.004	**0.730 ± 0.007**
CRNN-SED-1	0.494 ± 0.005	0.655 ± 0.005
CRNN-SED-2	**0.508 ± 0.003**	0.693 ± 0.007

The total number of parameters and the amount of computation (i.e., MACs) for the different systems are shown in Table 2. CCRN-SED-2 has the smallest Params, while official Baseline has the lowest MACs. The reason of CCRN having high MACs with low Params is that the complex module contains the structure of parameter sharing, such as a real CNN or image CNN that needs to operate twice with both real input and image input.

Tabel 2: Complexity of different systems

System	Params	MACs
Baseline	1.8M	**1.0360G**
CRNN-SED-1	1.4M	20.822G
CRNN-SED-2	**1.1M**	20.730G

4. CONCLUSION

In this report, we described our systems used in task 4 of DCASE 2024. The system is mainly based on CCRN, which can process both amplitude and phase information. We try to explore whether complex information can help the system to perform the sound event detection task better. Besides, the system uses BEATs pretrained model and Mean Teacher training strategy. The PSDS1 and mean pAUC of the final system on the development test set are 0.508 and 0.693.

5. REFERENCES

[1] Hu Y, Liu Y, Lv S, et al. DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement[J]. arXiv preprint arXiv:2008.00264, 2020.

[2] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in Proceedings of the 31st International Conference on Neural Information Processing Systems, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 1195–1204.

[3] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, “BEATs: Audio pre training with acoustic tokenizers,” arXiv preprint , arXiv:2212.09058, 2022.

[4] Serizel R, Turpault N, Shah A, et al. Sound event detection in synthetic domestic environments[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 86-90.

[5] Jiang Y, Guo D, Wang L, et al. Sound event detection in traffic scenes based on graph convolutional network to obtain multi-modal information[J]. Complex & Intelligent Systems, 2024: 1-16.

[6] Mnasri Z, Rovetta S, Masulli F. Anomalous sound event detection: A survey of machine learning based methods and applications[J]. Multimedia Tools and Applications, 2022, 81(4): 5537-5586.

[7] Cakır E, Parascandolo G, Heittola T, et al. Convolutional recurrent neural networks for polyphonic sound event detection[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017, 25(6): 1291-1303.

[8] Ma J, Wang R, Ji W, et al. Relational recurrent neural networks for polyphonic sound event detection[J]. Multimedia Tools and Applications, 2019, 78: 29509-29527.

[9] Li Y, Liu M, Drossos K, et al. Sound event detection via dilated convolutional recurrent neural networks[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 286-290.

[10] Yan J, Song Y, Dai L R, et al. Task-aware mean teacher method for large scale weakly labeled semi-supervised sound event detection[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 326-330.

[11] Serizel R, Turpault N, Shah A, et al. Sound event detection in synthetic domestic environments[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 86-90.

[12] Martín-Morató I, Mesaros A. Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation[J]. IEEE/ACM transactions on audio, speech, and language processing, 2023, 31: 902-914.

[13] Cornell, Samuele et al. “DCASE 2024 Task 4: Sound Event Detection with Heterogeneous Data and Missing Labels.” (2024).