SPECTRUM COMBINATION AND CONVOLUTIONAL RECURRENT NEURAL NETWORKS FOR JOINT LOCALIZATION AND DETECTION OF SOUND EVENTS

Technical Report

Shuangran Lueng

liangsr@outlook.com

Yi Ren

yi.ren@dbsonics.net

ABSTRACT

In this work, we combine existing Short Time Fourier Transforms (STFT) of 4-channel array audio signals to create new features, and show that this augmented input improves the performance of DCASE2019 task 3 baseline system [1] in both sound event detection (SED) and direction-of-arrival (DOA) estimation. Techniques like ensembling and finetuning with masked DOA output are also applied and shown to further improve both SED and DOA accuracy.

Index Terms— DCASE 2019 task 3, sound event detection, sound source localization, directions-of-arrival, multi-task learning

1. INTRODUCTION

Sound event detection (SED) and direction-of-arrival (DOA) estimation are two closely related tasks in multi-channel audio signal recognition, which are recently modelled jointly with neural networks [1, 2]. Specifically, the DCASE2019 task3 baseline system [1] proposed to use the log-magnitude and phase information of STFT of each audio channel as input to the neural net, whereas inter-channel information such as cross-correlation and auto-correlation matrix are commonly used in traditional localization methods like MUSIC and steering response power (SRP). Motivated by this, we propose in this work to stack new input channels corresponding to inter-channel information upon the ones used by [1], and show that such features improve the SED and DOA performance of the baseline to some extent.

The remaining of the paper is structured as follows: section 2 includes an overview of our system, where the preprossessing, network architecture and ensemble method are described in detail. Section 3 shows the experiment results on the development dataset, and how the results of different models are ensembled and submitted for evaluation. Section 4 concludes this work with some discussion.

2. SYSTEM OVERVIEW

2.1. Audio pre-processing

Our model takes as input the STFT spectra of the 4 input channels from the FOA dataset, as in the baseline system (section 2.1.1). Furthermore, we combine these spectra to generate more input channels representing the correlation between channels (section 2.1.2).

2.1.1. STFT

Each channel of the audio recordings are applied a STFT with 40ms window length (1920 samples) and 20ms hop length (960 samples). Then the complex spectra are normalized on a per-frequency basis, and denoted as

$$X_c(t,f) = X_{R,c}(t,f) + i \cdot X_{I,c}(t,f)$$

where $X_{R,c} \in \mathbb{R}^{1024\times T}$ and $X_{I,c} \in \mathbb{R}^{1024\times T}$ are the real and imaginary parts of the spectrum of channel $c \in \{1,2,3,4\}$ (The 0-frequency band is discarded).

2.1.2. Spectrum Combination

In this part, we define five groups of features in terms of $X_{R,c}$ and $X_{I,c}$, which are then combined to form different feature sets. First, we use the log-magnitude and angle of STFT as in the baseline system:

$$\log(X_{R,c}^2 + X_{I,c}^2) \tag{1}$$

$$\arctan_2(X_{I,c}, X_{R,c}) \tag{2}$$

Next, the real and imaginary parts

$$X_{I,c}, X_{R,c} \tag{3}$$

themselves also consist of a group of features. Although they contain the same information as (1) and (2), they are different facets of the input signal, and should be helpful to the modelling.

Finally, we add cross power spectrum between channels to the input to aid the localization task, as such information is commonly used in traditional localization methods (e.g. MUSIC and SRP). For example, the cross power spectrum between channels 1 and 2 can be expressed in frequency domain as:

$$\begin{aligned}\overline{X_1}X_2 &= (X_{R,1} - i\cdot X_{C,1})(X_{R,2} + i\cdot X_{C,2}) \\ &= (X_{R,1}X_{R,2} + X_{C,1}X_{C,2}) + i\cdot(X_{R,1}X_{C,2} - X_{C,1}X_{R,2})\end{aligned}$$

Therefore, we can add $X_{R,1}X_{R,2} + X_{C,1}X_{C,2}$ and $X_{R,1}X_{C,2} - X_{C,1}X_{R,2}$ to our input. However, we believe that given $X_{R,1}X_{R,2}, X_{C,1}X_{C,2}, X_{R,1}X_{C,2}, X_{C,1}X_{R,2}$ as inputs, neural networks are powerful enough to learn linear combinations of these second-order terms to form cross power spectrum and other useful features for localization. So as a generalization of cross power spectrum, we add all the second-order terms to the input:

$$X_{a_1,c_1} \cdot X_{a_2,c_2} \tag{4}$$

where $a_1, a_2 \in \{R, I\}, c_1, c_2 \in \{1,2,3,4\}, c_1 \le c_2$. In our experiments, we indeed observe that introducing these second-order terms leads to better result.

In summary, the groups (1), (2), (3), (4) have 4, 4, 8, 36 channels respectively, and three combinations of these groups are used as input in our experiments: {(2), (4)} (named c40 as it contains 40 channels), {(1), (2), (4)} (named c44) and {(2), (3), (4)} (named c48).

2.2. Network

We choose to use baseline CRNN in [1], while the only difference is that the input is changed from the original 8-channel one of the following: c40, c44 and c48 (all channels sliced over time to size 1024×128), and batch normalization (BN) is applied to the input before the convolutional layers.

2.3. Ensemble

In this work, we simply use weighted averaging of the model outputs for ensembling. First the 4 folds of each model are averaged with equal weights, then the fold averages of each model are averaged with different weights. Table 2 shows the the weights of each model and the performance of the ensemble systems. Note that SED and DOA results are ensembled independently with different weights.

3. EXPERIMENTS

3.1. Dataset

We use TAU Spatial Sound Events 2019 - Ambisonic dataset[3]. The recordings in the development and the evaluation sets are sampled at 48kHz, each recording lasting about 1 minute. The development set consists of 400 recordings, evenly divided into four cross-validation splits. The evaluation set consists of 100 recordings without labels.

3.2. Training

We use Adam optimizer with 0.01 learning rate and batch size 16 for training 80 epochs, early stopping is also adopted if no improvement of validation loss persists for 5 epochs. To avoid overfitting, dropout is applied with rate 0.3.

As can be seen in Table 1, the resulting model has better SED performance compared to the baseline, whereas the DOA error is not as satisfactory. To further decrease this error, we propose to finetune the DOA branch of the network (that is, the two time-distributed dense layers before the DOA output), while freezing the rest of the model. Furthermore, we notice that the DOA error is computed only within the SED events, however the DOA loss is computed over all classes and all time frames, which makes it unnecessarily harder to train. To alleviate this difficulty, we apply a masking layer to the original DOA output of the network, which performs the following operation:

$$y_d' = y_d \cdot P_s + y_0 \cdot (1 - P_s) \tag{5}$$

where y_d is the DOA prediction from the original model, P_s is the SED prediction, y_0 is a constant array representing the default DOA values in absence of sound events (azimuth 180° and elevation 50°), and y_d' is the final DOA prediction. In the finetuning stage, P_s is presumably highly accurate, so we exploit it to indicate the active sound events. High P_s values imply present events, whose the DOA errors shall be taken into account; On the other hand, low P_s values means the corresponding events are abscent, and should be ignored in the loss computation (therefore their DOA values are replaced by the defaults).

The finetuning is achieved with a learning rate of 0.0002, 0.01 decay and a cyclic learning rate scheduler. As shown in Table 1, finetuning effectively reduces the DOA error.

As a side note, our masking layer approach is somehow similar to [2], where they use the ground truth SED as the mask for DOA during training.

3.3. Evaluation

SED and DOA estimation are evaluated separately, using the following frame-based (each frame lasts 20ms) evaluation metrics: error rate (ER) and f-score for SED[4], DOA error and frame recall for DOA estimation[5]. Predictions from all folds are evaluated as a single experiment to avoid biases under cross-validation[6].

Single system results for different feature sets, with or without the second stage finetuning are listed in Table 1. In this table, we see that among different feature sets, c40 (no angles information provided) is less accurate than c44 and c48. In addition, for a fixed a feature set, DOA estimations are in general better with finetuning in a small sacrifice of SED performance.

Table 1: Evaluation metrics for single system results

Features combination	Finetune	SED ER	SED F-score	DOA ERROR	DOA FRAME-RECALL
Baseline	-	0.34	79.9%	28.5°	85.4%
c40	no	0.241	85.8%	35.1°	88.0%
c44	no	0.238	86.4%	27.4°	88.4%
c44	yes	0.246	86.2%	26.6°	88.7%
c48	no	0.231	86.4%	28.6°	88.2%
c48	yes	0.237	86.6%	27.4°	88.7%

Ensemble systems have better performance in all the four metrics, as listed in table 2.

Table 2: Evaluation metrics for ensemble systems

Ensemble system	SED ER	SED F-score	DOA ERROR	DOA FRAME-RECALL
s1	0.201	88.1%	26.9°	89.0%
s2	0.197	88.4%	25.4°	89.6%

System s1 is an emsemble of models trained with feature c40, c44 and c48 without finetuning, with ensembling weights being [0.33, 0.33, 0.33] for SED and [0.048, 0.476, 0.476] for DOA estimation. System s2 is an ensemble of models trained with feature c40, c44, c48 without finetuning, and two finetuned models with feature c44, c48, ensembling weights being [0.2, 0.2, 0.2, 0.2, 0.2] for SED and [0., 0.25, 0.25, 0.25, 0.25] for DOA estimation.

3.4. Submission

We submitted two systems: `Leung_DBS_task3_1` and `Leung_DBS_task3_2`, corrsponding to s1 and s2 in Table 2.

4. CONCLUSION AND DISCUSSION

In this work, we see that by extending the inputs with new channels, the SED and DOA estimation performances are improved. Furthermore, this method can potentially give any network performance

a boost as it is general and independent to the network architecture. Besides, we observe that ensembling and finetuning with DOA mask layer are also beneficial.

On the side note, we also experimented on an augmented dataset containing remixed audios with an increased number of polyphonic events, but the model suffered from severe underfitting. It's left for future work to detect polyphonic events more accurately.

5. REFERENCES

[1] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, 2018.

[2] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. D. Plumbley, "Polyphonic sound event detection and localization using a two-stage strategy," 2019.

[3] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and uetection," in *Submitted to Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019. [Online]. Available: https://arxiv.org/abs/1905.08546

[4] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/162

[5] S. Adavanne, A. Politis, and T. Virtanen, "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1462–1466.

[6] G. Forman and M. Scholz, "Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement," *SIGKDD Explor. Newsl.*, vol. 12, no. 1, pp. 49–57, November 2010. [Online]. Available: http://doi.acm.org/10.1145/1882471.1882479