SOUND EVENT DETECTION FROM REAL-LIFE AUDIO BY TRAINING A LONG SHORT-TERM MEMORY NETWORK WITH MONO AND STEREO FEATURES

Chun-Hao Wang, Jun-Kai You, Yi-Wen Liu

Dept. Electrical Engineering, National Tsing Hua University, Hsinchu City, Taiwan

ABSTRACT

In this paper, we trained and evaluated an acoustic sound event classifier that uses a combination of stereo and mono features. For stereo features, we treated the time difference of arrival (TDOA) as a random variable and calculated its probability density function. For mono features, Mel-frequency cepstral coefficients (MFCCs) and their 1st and 2nd derivatives were extracted. A recurrent neural network (RNN) with long-short term memory (LSTM) was constructed to perform multi-label classification. Training with the 4-fold validation dataset given by DCASE2017 challenge [5], model parameters were chosen based on the best average performance. The proposed TDOA plus MFCC features combined with the RNN-LSTM model achieved a segment-based error rate of 0.77. In DCASE2017 challenge, the proposed model gets segment-based error rate of 0.9749 and F-score of 40.8% in overall evaluation dataset.

Index Terms— Sound Event Detection, Recurrent Neural Network, Long-Short Term Memory, Time Difference of Arrival

1. INTRODUCTION

Sound event detection (SED) attracts attention in recent years for home applications. In real life, sound events happen everywhere, anytime, and can mix with one another. A SED system has to deal with overlapping events under noisy acoustic conditions. Thus, it is challenging to design a SED system that can accurately estimate event onset and offset time. Existing methods include non-negative matrix factorization (NMF) [1][2], neural networks (NNs) [3][4], and so on. NMF based methods intrinsically allow events to overlap in time. NNs, capable of automatic extraction of signal representations from the input data in a flexible manner, have exhibited excellent performance in classification tasks across different application domains; in [3], a deep neural network (DNN) with fully-connected hidden layers was built to take concatenated features as its input and solve the time dependency problems. In [4], an RNN-LSTM structure exhibited capability to handle sequential data, and achieved a high performance in DCASE2016.

Inspired by [4], we decided to choose RNN-LSTM as our classification model while using stereo and mono input features. Post-processing techniques are also considered. In brief, two sets of features are extracted --- the first set contains 20 Mel-frequency cepstral coefficients (MFCC) and their 1st and 2nd derivatives; the second set describes a probability density function (PDF) of time difference of arrival (TDOA), or equivalently the sound direction of arrival (DOA), sampled at a number of different latencies (or equivalently arrival angles [8] under the far-field assumption). An RNN-LSTM model is trained by these features. Afterwards, the output labels are subject to smoothing. The rest of this paper is organized as follows: Sec. 2 and 3 describe the feature extraction methods and the learning methodology, respectively. Sec. 4 reports and discusses the event detection and classification performance of the present system on DCASE2017 database. Sec. 5 gives the conclusions.

2. FEATURE EXTRACTION

This section describes feature extraction in details.

2.1. MFCC Settings

MFCC is widely used in speech recognition and audio processing. It partitions the audible frequency range into non-uniform bandwidths based on human auditory perception. In this research, MFCC is extracted by following configuration: frame length = 40ms, 50% overlap, Hann windowing, 40 Mel-filters, and 20 discrete cosine transform coefficients. The 1st and 2nd derivatives with respect to time are also calculated. We retain the log energy term in MFCC to reflect the difference in sound volume between different kinds of sound events. Finally, z-normalization is performed so each feature has a zero mean and standard deviation of one.

2.2. TDOA

Humans are known to be able to distinguish different sound sources based on binaural cues. For low frequency components in particular, the human auditory system is known to be able to perform neuro-biological calculation of the direction of arrival based on interaural time difference (ITD) [7]. By informally listening to the materials given by this year's DCASE via a headphone, we found that some events could be perceived binaurally as if the direction of arrival is changing. Therefore, we attempt to imitate the human auditory system by jointly considering the signals received by two channels so as to estimate the time difference of arrival (TDOA) of each event. We are interested in seeing if the event classification accuracy could be improved by including TDOA-based features.

The TDOA is defined as follows,

$$\Delta t = t_R - t_L, \tag{1}$$

where t_R and t_L denote the time it takes for a sound to propagate from the source location to the right and left channel, respectively. If the source moves from the recording pair of microphones' left to their right, the TDOA would decrease. We adopted a tech-

nique to estimate TDOA via probabilistic modeling [8]. The probability density function of TDOA is written as follows:

$$p_{TDOA}(\Delta t) = \prod_{n=1}^{N/2-1} p_{IPD,n}(2\pi n f_s \Delta t / N \ mod\ 2\pi) \quad (2)$$

where $p_{IPD,n}$ is PDF of inter-channel phase difference (IPD) between left and right, n is the frequency index, f_s denotes the sampling frequency, and N denotes the length of FFT. Assuming that the distance d_{mic} between two microphones is known, the time difference Δt should be limited within the range $[-\frac{d_{\text{mic}}}{c}, \frac{d_{\text{mic}}}{c}]$, where c denotes speed of sound. Figure 1 shows a typical variation of the PDF against time when a car passes by. The brightness indicates high probability density. The audio clip was selected from in this year's DCASE database. By inspection, we can infer that the car sound recorded in this clip moved to microphones' right hand side because TDOA decreases from approximately 1.6 to 2.5 seconds. Empirically, we actually perceived that the car moves to right via the headphone.

Hence, a maximum likelihood estimation of TDOA can be obtained as follows,

$$\Delta\tilde{t} = arg\max_{\Delta t \in [-\frac{d_{mic}}{c}, \frac{d_{mic}}{c}]} \ p_{TDOA}(\Delta t), \quad (3)$$

Here, $\Delta\tilde{t}$ can be used as one single feature that varies from one frame to the next. In addition to using $\Delta\tilde{t}$, it is possible to use the entire PDF of TDOA as a feature vector. In this work, we sample the range of possible TDOAs at 65 equally spaced points. In principle, the TDOA vector can be concatenated with MFCCs (and its derivatives) directly to train the RNN-LSTM model. In practice, we subtract the feature vector by its maximum for each frame so the peak value is always 0.

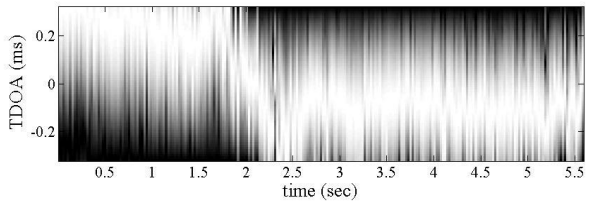

Figure 1: variation of $p_{TDOA}(\Delta t)$ against time for a "car" event in the DCASE2017 database

3. SYSTEM DESCIPTION

This section describes the details of system configurations and training methods. The system diagram is shown in Fig. 2.

3.1. System Overview

We apply short-time feature extraction mentioned in Sec. 2 and train a RNN-LSTM sound event classifier. When training a RNN-LSTM model, the best combination of the following parameters is searched exhaustively: the learning rate, the number of timesteps in RNN-LSTM, and the batch size. Obeying the official instruction to perform 4-fold training, we obtain four models with different combinations of parameters. The best training parameters are determined based on the average validation loss.

Afterwards, we use the best parameters to train the RNN-classifier with all the training data and predict the occurrence of sound events in the official DCASE testing dataset. The prediction is subject to post-processing (to be described in 3.4) to produce reliable event onset and offset time.

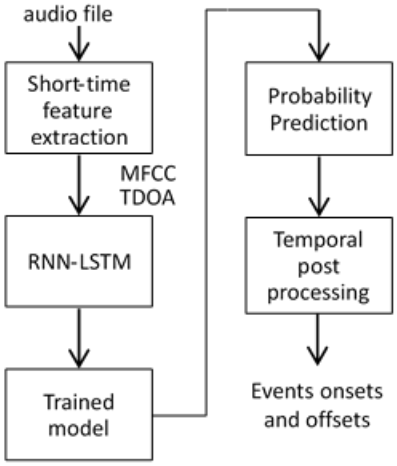

Figure 2: System overview

3.2. RNN-LSTM

RNNs differ from traditional neural networks in one major way--in traditional neural networks, data are passed forward, and weights are modified via the back propagation algorithm. Samples in the database are regarded as temporally independent. RNNs, in contrast, consider time dependency. The output of RNNs depends upon the present data and previous data. Therefore, RNNs are supposed to handle data that are acquired in time sequentially. The RNN structure had suffered from the vanishing and exploding descent problem in the past, but in recent years the problems have been solved by the usage of long-short term memory (LSTM) [9], among other solutions as well.

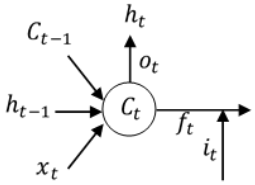

Figure 3: RNN-LSTM architecture

The RNN-LSTM architecture is depicted in Fig. 3. The main equations for its operation are listed below [9][11],

$$\begin{aligned} i_t &= \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \\ f_t &= \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \\ o_t &= \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \\ c_t &= f_t \otimes c_{t-1} + i_t \otimes \tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c) \\ h_t &= o_t \otimes \tanh(c_t) \end{aligned} \quad (4)$$

where i_t, f_t and o_t represent input gate, forget gate, and output gate respectively, c_t and h_t are memory cell and cell output in present; in Eq. (4), the W terms (with different subscripts) are weight matrices, b terms are the biases, and σ is an activation function, $\otimes$ is element-wise product operator.

What makes LSTM unique is the usage of the gates. The gates are controlled by the activation function so they produce output within a fixed range. The memory cell c_t receives previous memory cell c_{t-1}, previous output h_{t-1}, and present input x_t. The data go through input gate and the memory-cell output goes through forget gate. The next memory cell can preserve the characteristic of the previous data while responding to new input data at the same time. Thus it has the advantage to handle sequential data.

The system configuration was set as follows --- the input layer has the same number of units as the dimension of the feature vectors; the hidden part consists two LSTM layers each with 32 units; the final layer is fully-connected with 6 units representing all of the targeted sound events, for which the activation function is the sigmoid function and the loss function is the binary cross entropy. The RMSProp algorithm [10] is used for weight optimization. Occasionally, we adjust the model structure slightly while dealing with different features. The details will be described in section 4.3. As for model parameters, we tried out all combinations between the following parameter values: the learning rate at 0.00005, 0.0001, or 0.0005, the batch size of 32 or 64, and the number of timesteps at 50, 100, or 150. The best combination is thus determined based on average validation loss.

3.3. Thresholding

While the RNN-LSTM model is supposed to produce binary labels, the output of the final layer takes continuous values between 0 and 1. A threshold for event activation needs to be determined for every targeted event. For this purpose, we treat each event as a different classification task, and exhaustively search for the best threshold value between 0.2 and 0.8 in steps of 0.012. Error rates are recorded in a fold-wise manner, and the best threshold for each targeted event is determined so as to minimize the error rate across four folds defined in the official DCASE2017 instruction.

3.4. Post-processing

Heuristic post-processing techniques are applied to reduce overly frequent switching between event onsets and offsets. First, activations are merged if the gap in between is shorter than 150 frames (or equivalently, 3.0 second). Then, we calculate the distribution of the duration for each targeted event class and remove an event from the final list if its duration is shorter than 100 frames (or equivalently, 2.0 second).

4. EVALUATION

4.1. Dataset and Metrics

TUT Sound Events 2017 dataset [5] was made available for DCASE2017 challenge task 3. The whole dataset for DCASE2017 task 3 consists of real audio recordings from the street scene, and it was divided into a development part and an evaluation part. All recordings are stereo and were digitized at a sampling rate of 44.1 kHz with 24-bit resolution. Target events consist of 6 classes: brakes squeaking, car, children, large vehicle, people speaking, and people walking. Label for event onsets and offsets are provided for the development part of the database so supervised learning can be conducted.

An evaluation metrics [6] for task 3 is also defined precisely; the performance will be judged by segment-based error rate, and the length of the segments is 1.0 second.

4.2. Baseline

The DCASE2017 challenge baseline system [5] uses a multi-layer perceptron (MLP) structure with the log mel-band energy as the input features. The feature is also calculated every 20ms with a 50% overlap between adjacent frames. The MLP classifier takes five consecutive frames into consideration at once so as to handle time-dependency. The baseline system achieves a segment-based error rate of 0.69 and an F-score of 56.7% by testing across all four of the officially defined folds.

4.3. Features and structures

Table 1 summarizes the various structures we have tuned and evaluated in experiments. Table 2 summarizes the features we use for training the system.

	Description
Architecture 1 ($arch_1$)	Input -> 2 hidden layer (32 units) -> 1 fully-connected output layer (6 units)
Architecture 2 ($arch_2$)	Input->1 fully-connected layer (39 units) -> 2 hidden layer (32 units) -> 1 fully-connected output layer (6 units)
Architecture 3 ($arch_3$)	Input1 -> 2 hidden layer (32 units) -> M1 Input2 -> 2 hidden layer (32 units) -> M1 M1 -> 1 fully-connected output layer (6 units)
Architecture 4 ($arch_4$)	Input1 -> 2 hidden layer (13 units, 1 unit) -> M1 Input2 -> 1 hidden layer (1 unit) -> M1 M1 -> 1 fully-connected output layer (6 units)

Table 1: RNN-LSTM structure description. Arch1 and arch2 are single training lines. Arch3 and arch4 start with 2 training lines and combine them into the merging model M1.

Features	**Description**
MFCC_{39}	40 Mel-filters, 20 DCT, 13 MFCCs, delta and acceleration
MFCC_{60}	40 Mel-filters, 20 DCT, 20 MFCCs, delta and acceleration
TDOA_{65}	65 log probabilities of TDOA
TDOA_{1}	1 maximum-likelihood TDOA for each frame and processing with median filter of length 5.

Table 2: Features used for training. The subscripted number

attached to the feature name denotes the dimension of the feature vectors.

4.4. Performance of the methods

Table 3 summarizes the performance of different combinations of features and architectures. The results show that MFCC_{60}-based combinations achieve a segment-based error rate of 0.83. The combination of MFCC features and TDOA features with $arch_3$ reaches the best performance with a segment-based error rate of 0.77. The present results suggest that it helps to include TDOA-based features in terms of segment-based recognition accuracy. However, by examining the error rates for different classes of sounds, we found that the "car" recognition error rate in MFCC_{60}+TDOA_{65}+$arch_3$ is significantly lower compared to others classes. We also noticed that the "car" event seems to outnumber other events in this dataset of street sounds. Therefore, we argue that the inclusion of TDOA might not help so much in other scenes if none of the target events originate from fast-moving sound sources. For this dataset, note that even using TDOA65 alone achieves an error rate of 0.85; this performance is not much worse than using MFCCs.

Models	Raw	Post-processing
Baseline	-	0.69
MFCC_{60}+$arch_1$	0.86	0.83
MFCC_{60}+$arch_2$	0.91	0.85
TDOA_{65}+$arch_1$	0.88	0.85
MFCC_{39}+$arch_1$	0.92	0.80*
MFCC_{39}+TDOA_1+ $arch_4$	1.5	1.49
MFCC_{60}+TDOA_{65}+ $arch_3$	0.81	**0.77**

Table 3: Segment-based error rates achieved by various combinations of features and architectures. (*: For MFCC_{39}+$arch_1$, the performance 0.80 was obtained by averaging over only three folds because we encountered an erroneous situation in which no event was detected in fold #4).

4.5. Performance in Evaluation data

Evaluation dataset contains 8 recordings with 2~5 minutes as the same configurations as section 4.1. We will evaluate the model performance by calculating segment-based error rate, and F-score is calculated as well in evaluation dataset. The RNN-LSTM model with $arch_3$ achieves the segment-based error rate of 0.9749, and F-score of 40.8%. Table 4 shows the class-wise performance. The "car" event recognition error rate is 0.8315. It is still much better than other events. The "brakes squeaking" event gets the error rate of 1 means that the event is not detected by the model in all evaluation dataset. The rest of the class events get the error rate over 1.

Target event	Error rate
Brakes squeaking	1
Car	0.8315
Children	2.4222
Large vehicle	2.0678
People speaking	1.6367
People walking	1.3094

Table 4: Class-wise segment-based error rates calculated in evaluation dataset.

5. CONCLUSION

Through the experiments we can see that, in terms of lowering the segment-based error rates, it helps to include both the stereo (TDOA) features and the mono (MFCC) features. The RNN-LSTM model reached an error rate of 0.77, better than using only the TDOA features (0.85) or using only the MFCC features (0.83) at their best. Even though the present performance in terms of segment-based error rate is inferior to the baseline (0.69), we would suggest that taking multiple channels of recordings into account is generally a good practice when attempting to detect and classify acoustic events, especially when some of the targeted sounds might originate from moving sources. These being said, we submitted our results for DCASE2017 challenge using the present parameters, features (MFCC60 + TDOA65), and architecture ($arch_3$). In DCASE2017 challenge, the model achieves segment-based error rate of 0.9749 in overall evaluation dataset.

6. REFERENCES

[1] Sobieraj, I., & Plumbley, M. (2016). "Coupled Sparse NMF vs. Random Forest Classification for Real Life Acoustic Event Detection," In *Proc. Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE 2016)*, pp. 90-94.

[2] Giannoulis, P., Potamianos, G., Maragos, P., & Katsamanis, A. (2016). "Improved dictionary selection and detection schemes in sparse-CNMF-based overlapping acoustic event detection," *Proc. DCASE 2016*.

[3] Kong, Q., Sobieraj, I., Wang, W., & Plumbley, M. D. (2016). "Deep neural network baseline for DCASE challenge 2016," *Proc. DCASE 2016*.

[4] Adavanne, S., Parascandolo, G., Pertilä, P., Heittola, T., & Virtanen, T. (2017). "Sound event detection in multichannel audio using spatial and harmonic features," *arXiv preprint arXiv:1706.02293*.

[5] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen. (2017). "DCASE 2017 challenge setup: tasks, datasets and baseline system," in *Proc. DCASE2017*. November 2017, submitted.

[6] Mesaros, A., Heittola, T., & Virtanen, T. (2016). "Metrics for polyphonic sound event detection," *Applied Sciences*, **6**(6), p. 162.

[7] Strutt, J. W. (1907). "On our perception of sound direction," *Philosophical Magazine*, vol. 13, p. 214.

[8] Li, C.-W. and Liu, Y.-W. (2016). "Posterior probabilistic modeling for inter-channel phase and time difference estimation in audio signals," *Proc. IEEE ICASSP*, Shanghai, China, March 2016.

[9] Hochreiter, S., & Schmidhuber, J. (1997). "Long short-term memory," *Neural Computation*, **9**(8), 1735-1780.

[10] Tieleman, T., & Hinton, G. (2012). "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," from *COURSERA: Neural Networks for Machine Learning*, **4**(2), 26-31.

[11] Gers, F. A., Schmidhuber, J., and Cummins, F., (1999). "Learning to forget: Continual prediction with LSTM," *Neural Computation*, **12**(10), pp. 2451-2471.