MULTI-SCALE RECURRENT NEURAL NETWORK FOR SOUND EVENT DETECTION

Rui Lu[1] *Zhiyao Duan*[2*] *Changshui Zhang*[1†]

[1] Department of Automation, Tsinghua University
State Key Lab of Intelligent Technologies and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, P.R.China
[2]Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA

ABSTRACT

Sound event detection (SED) in real life is an interesting but challenging task due to the polyphonic and long-term dependent nature of sound events. Recently, multi-label recurrent neural networks (RNNs) have shown promises. However, even equipped with long short-term memory (LSTM) or gated recurrent unit (GRU) cells, RNNs are still limited to model the long-term dependency. In this paper, we propose a multi-scale RNN to address this issue. By integrating information from different time resolutions, we can better capture both the fine-grained and long-term dependencies of sound events. We experiment on the development sets of Task3 of DCASE2016 and DCASE2017. Compared to our previously proposed single-scale RNN that won the third place among the 13 teams in Task3 of DCASE2017, the proposed multi-scale model achieves statistically significantly better performance on the development datasets of both DECASE2016 and DCASE2017.

Index Terms— Multi-scale model, deep learning, recurrent neural network, sound event detection

1. INTRODUCTION

Sound event detection (SED), i.e., the automatic detection of environmental sounds and their temporal boundaries, has been increasingly popular in recent years because of its wide applications in machine hearing, scene understanding and surveillance. Sounds convey important information about our surroundings, and supplement information obtained from other modalities such as cameras and motion sensors [1].

There are several challenges in SED. First, environmental sound scenes are naturally polyphonic since sound sources are rarely heard in isolation in everyday life [2]. The concurrency of multiple sound sources (i.e., *polyphony*) enriches diversity of the acoustic scene, and makes the detection more difficult than that in monophonic sound recordings [3]. How to design richer models to capture this diversity is one challenge. Second, temporal dependency is a key discriminative aspect of sound events. There are generally two types of temporal dependencies: fine-grained and long-term dependency. Fine-grained dependency is at the scale of miliseconds and captures the subtle fluctuation of the frequency content. Long-term dependency is at a coarser scale and its length depends on the nature and length of the event, it is typically at the scale of seconds [4]. To model the fine-grained dependency, one typically uses a frame length and hop size at the scale of tens of miliseconds, but this means that the long-term dependency would cover hundreds of frames, which is difficult to model. One naive approach is to use a longer frame length and hop size, but this would miss the fine-grained dependency. How to model the long-term dependency without sacrificing the fine-grained dependency is another challenge.

Early systems for SED include hidden Markov models (HMMs) [2] and non-negative matrix factorization (NMF) [5]. Recently, recurrent neural networks (RNNs) equipped with long short-term memory (LSTM) cells [3] and convolutional neural networks (CNNs) [6] are adopted and have outperformed early systems. However, their capability for modeling the long-term dependency is still quite limited: the maximum length of dependency that LSTM can well capture is between 30 and 80 [7], one order of magnitude smaller than the long-term dependency typically spans in sound events.

One way to model the long-term dependency without sacrificing the fine-grained dependency is to use multi-scale models: different levels of the multi-scale structure model the data sequence at different time scales [8]. Multi-scale models have been adopted for music auto-tagging [9] and music emotion prediction [10]. However, these tasks only require a single prediction for a given audio recording. In the SED task, we need to continuously output labels in each audio frame, based on which the temporal boundaries of the overlapping sound events are estimated.

In this paper, we propose a multi-scale RNN to balance the modeling of both the fine-grained and long-term dependency: it uses one bidirectional RNN to model the audio input at the resolution of 0.02 seconds (audio frame hop size) with a time span of 0.5 seconds, and uses another bidirectional

*ZD acknowledges the National Science Foundation Grant No. 1617107.
†CZ acknowledges the German Research Foundation (DFG) in Project Crossmodal Learning DFC TRR-169.

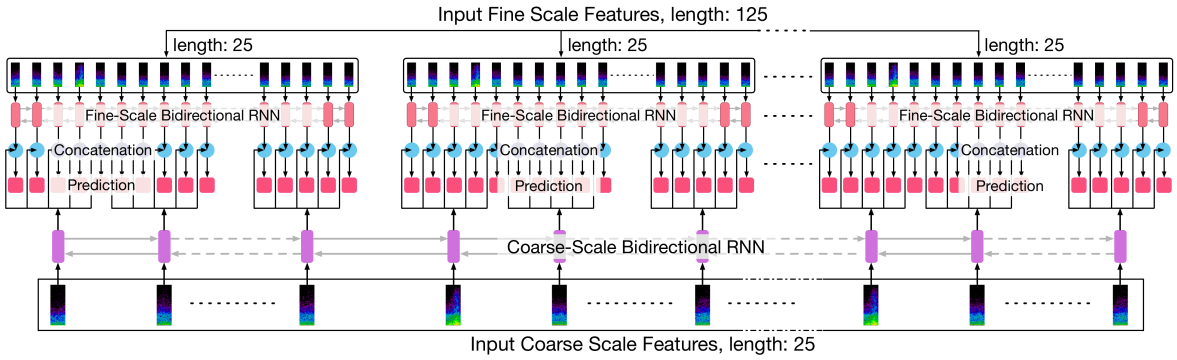

Fig. 1. Multi-scale RNN: RNNs on both fine- and coarse- scale are shown as rectangles. Circles in the middle represent concatenation operations while squares indicate the final single-layer fully connected network for sound event prediction on the fine-scale. Input fine-scale sequence has a length of 125 and coarse-scale with 25. During computation, the fine-scale sequence is split into five subsequences to feed into five fine-scale RNNs respectively.

RNN to model the input at the resolution of 0.1 seconds with a time span of 2.5 seconds. Both RNNs process sequences with a length of 25 steps, but with different resolutions and time spans. Outputs of the RNNs are concatenated at the frame level and then passed through a fully connected network to predict sound events. Compared to the traditional single-scale RNNs, this multi-scale architecture reduces the length of sequences over which the RNNs must reason to model long-term dependency. We further improve the performance of the proposed multi-scale RNN model by data augmentation [11] and ensemble learning. We experiment on the development sets of Task 3 of DCASE2016 and DCASE2017 following the official setup [12, 13]. Compared to a state-of-the-art single-scale RNN, which won the third place among the 13 teams in DCASE2017 Task 3, our proposed multi-scale RNN reduces the error rate by a significant amount on the development set.

2. METHOD

2.1. The Proposed Multi-Scale RNN

We propose a multi-scale RNN to better model the temporal dependencies of sound events: it comprises two RNNs which separately work at the fine-scale (f-RNN) and the coarse-scale (c-RNN). Both RNNs are bidirectional to exploit the contextual information from both time directions. Both RNNs contain multiple layers, where the output of the previous recurrent layer is fed as input to the next recurrent layer and the last recurrent layer is used to make the prediction. The two RNNs are aligned so that each cell of the c-RNN interacts with five cells of the f-RNN. Our model is shown in Fig. 1.

Suppose we are given a sequence of fine- and coarse-scale input vectors: $\{\mathbf{x}^t\}$ and $\{\mathbf{X}^T\}$. The f-RNN has N_1 layers and its hidden activations at the n-th layer for both directions are denoted as $\{\overrightarrow{\mathbf{h}}_n^t\}$, $\{\overleftarrow{\mathbf{h}}_n^t\}$, respectively. Similarly, the c-RNN has N_2 layers and the hidden activations at the n-layer for both directions are $\{\overrightarrow{\mathbf{H}}_n^T\}$ and $\{\overleftarrow{\mathbf{H}}_n^T\}$, respectively. For each recurrent layer, the recurrent process can be described as:

$$
\begin{aligned}
[\overrightarrow{\mathbf{h}}_n^t, \overleftarrow{\mathbf{h}}_n^t] &= \mathbf{fRNN}(\overrightarrow{\mathbf{h}}_{n-1}^t, \overleftarrow{\mathbf{h}}_{n-1}^t, \overrightarrow{\mathbf{h}}_n^{t-1}, \overleftarrow{\mathbf{h}}_n^{t+1}) \\
[\overrightarrow{\mathbf{H}}_n^T, \overleftarrow{\mathbf{H}}_n^T] &= \mathbf{cRNN}(\overrightarrow{\mathbf{H}}_{n-1}^T, \overleftarrow{\mathbf{H}}_{n-1}^T, \overrightarrow{\mathbf{H}}_n^{T-1}, \overleftarrow{\mathbf{H}}_n^{T+1}),
\end{aligned} \tag{1}
$$

where $t = 5(T-1) + i$ for $i \in \{1, 2, 3, 4, 5\}$. For the first recurrent layer (i.e., when $n = 1$), $[\overrightarrow{\mathbf{h}}_{n-1}^t, \overleftarrow{\mathbf{h}}_{n-1}^t]$ is replaced by $\mathbf{x}^t$, and $[\overrightarrow{\mathbf{H}}_{n-1}^T, \overleftarrow{\mathbf{H}}_{n-1}^T]$ is replaced by $\mathbf{X}^T$.

Final probabilistic predictions $\{\mathbf{y}^t\}$ are made at the fine-scale: at each fine-scale time step, we replicate the hidden states of the associated c-RNN cell and concatenate it with the hidden state of f-RNN cell, then pass it through a fully connected layer with sigmoid output as the following :

$$
\mathbf{y}^t = \sigma(\mathbf{W}[\overrightarrow{\mathbf{h}}_{N_1}^t, \overleftarrow{\mathbf{h}}_{N_1}^t, \overrightarrow{\mathbf{H}}_{N_2}^{\lceil \frac{t}{5} \rceil}, \overleftarrow{\mathbf{H}}_{N_2}^{\lceil \frac{t}{5} \rceil}] + \mathbf{b}), \tag{2}
$$

where $\mathbf{y}^t \in [0, 1]^L$, L is the number of all sound events, $\sigma(\cdot)$ is an element-wise sigmoid function, and $\lceil \cdot \rceil$ the ceiling operation. $\mathbf{W} \in \mathbb{R}^{L \times 2(d_f + d_c)}$ and $\mathbf{b} \in \mathbb{R}^L$, where d_f and d_c are the hidden sizes of f-RNN and c-RNN, respectively.

The training process of our proposed multi-scale RNN contains two stages: in the first stage, we train f-RNN and c-RNN separately with a sequence length of 25 and a hop size of 5. Their recurrent process are the same with (1), but the output is slightly different:

$$
\begin{aligned}
\mathbf{y}_f^t &= \sigma(\mathbf{W}_f[\overrightarrow{\mathbf{h}}_{N_1}^t, \overleftarrow{\mathbf{h}}_{N_1}^t] + \mathbf{b}_f) \\
\mathbf{y}_c^T &= \sigma(\mathbf{W}_c[\overrightarrow{\mathbf{H}}_{N_2}^T, \overleftarrow{\mathbf{H}}_{N_2}^T] + \mathbf{b}_c),
\end{aligned} \tag{3}
$$

where $\mathbf{W}_f \in \mathbb{R}^{L \times 2d_f}$, $\mathbf{b}_f \in \mathbf{R}^L$ and $\mathbf{W}_c \in \mathbb{R}^{L \times 2d_c}$, $\mathbf{b}_c \in \mathbf{R}^L$. During the second training stage, we feed the coarse-scale sequence with length of 25 and fine-scale with length of 125 simultaneously to the multi-scale RNN. We fix the parameters of both RNNs trained in stage 1, discard $[\mathbf{W}_f, \mathbf{b}_f, \mathbf{W}_c, \mathbf{b}_c]$ and train $\mathbf{W}$, $\mathbf{b}$ in (2) from scratch. We

constrain all sequences to the length of 25 to alleviate the gradient vanishing problem. Both training stages are supervised by minimizing the binary cross entropy (BCE) loss:

$$loss(\mathbf{y}^t, \hat{\mathbf{y}}^t) = -\frac{1}{L}\sum_{i=1}^{L}[\hat{y}_i^t log(y_i^t) + (1-\hat{y}_i^t) log(1-y_i^t)], \quad (4)$$

where $\hat{\mathbf{y}}^t \in \{0,1\}^L$ is sound events' ground-truth labels. Note that RNNs in our model can be equipped with LSTM or GRU, which will be discussed in the experimental setup.

2.2. Feature Extraction

We re-sample all files to 44.1 kHz and extract three sets of features (timbre, pitch, and TDOA) at two scales with hop sizes of 20 ms and 100 ms, denoted as the *fine-scale* and *coarse-scale*, respectively. For stereo recordings, timbre and pitch features are extracted on the average of the two channels.

Timbre Features: The first set of features are the log mel-spectrogram (lms) and mel-frequency cepstral coefficients ($mfcc$) [3, 14]. We first apply Hann window and STFT with a window length of 40 ms for the fine-scale and 93 ms for the coarse-scale. Then, we apply 40-band mel-scale triangular filters ranging from 0 to 22,050 Hz to obtain the 40-d lms feature. We transform the lms to the cepstral domain to extract the first 20-d $mfcc$ features and their first- and second-order derivatives with respect to time. We discard the first dimension of $mfcc$, resulting in a 59-d $mfcc$ feature.

Pitch Features: We detect the three most dominant frequencies within 100-4000 Hz at each scale, after taking STFT with window lengths of 46 ms (fine-scale) and 93 ms (coarse-scale), respectively [15]. Together with the corresponding periods, we extract a 6-d feature (denoted as *pitch3*) at each scale. We also extract a 2-d feature (denoted as *pitch*) by only keeping the single most dominant frequency and its period. These dominant frequencies are not necessarily fundamental frequencies (pitches); we use the name *pitch* for simplicity.

TDOA Features: The time difference of arrival (TDOA) at time t is defined as the time delay Δ that maximizes the correlation between the two channels at each mel-band [15]:

$$R_b(\Delta, t) = \sum_{k=0}^{N-1} H_b(k) \frac{X_1(k,t)X_2^*(k,t)}{|X_1(k,t)||X_2(k,t)|} e^{i2\pi k\Delta/N}, \quad (5)$$

where $H_b(k)$ is the magnitude frequency response of the b-th mel-band filter, $X(k,t)$ is the complex Fourier spectrum at frequency k and time t, $*$ is the conjugate operation, and N is the total number of frequency bins. We use five mel-bands ($B = 5$) ranging from 0 to 22,050 Hz and calculate FFT with three window lengths - 120, 240 and 480 ms for the fine-scale and 240, 360 and 600 ms for the coarse-scale. This results in a 15-d feature ($tdoa3$) at each scale. For each mel-band, we further take the median value of the TDOA from the three window lengths for noise reduction. This results in a 5-d feature ($tdoa$) at each scale.

We search for the best feature combinations to use for f-RNN and c-RNN independently. To do so, we treat each RNN as a stand-alone SED model and train and evaluate it for each feature combination for 10 times. We choose the combination that achieves the lowest error rate (ER) on the validation set. For DCASE2017, the best combination is $mfcc + pitch3$ for f-RNN and $mfcc + pitch$ for c-RNN. For DCASE2016, the best is $lms + pitch + tdoa$ for f-RNN and $mfcc + pitch3$ for c-RNN, where the lms and $pitch$ features for f-RNN are concatenations of the two channels instead of the average.

2.3. Data Augmentation and Ensemble

To address the issue of data scarcity, we augment the training data with the union of "pitch shift" and "time stretch" deformations [11]: for each recording, we tune the pitch by one of the 14 semitones: $\{\pm 0.5, \pm 1, \pm 1.5, \pm 2, \pm 2.5, \pm 3, \pm 3.5\}$ or change the speed by one of the 10 factors: {0.71, 0.76, 0.81, 0.87, 0.93, 1.07, 1.15, 1.23, 1.32, 1.41}. We also implement an ensemble technique: we train four models with the same setting except that the validation datasets are randomly selected from training set. During testing, frame-wise predictions of the four models are averaged before post-processing.

3. EXPERIMENTS

3.1. Setup

We experiment on the development datasets of Task 3 of DCASE2016 and DCASE2017 [4]. The DCASE2016 dataset contains 22 recordings in $home$ and $residential\ area$. The $home$ context has 10 recordings with 11 sound events while the $residential\ area$ context has 12 recordings with 7 sound events. The DCASE2017 dataset consists of 24 recordings in the $street$ context with 6 types of sound events. All of the recordings have a length of three to five minutes.

For DCASE2016, we use a two-layer bidirectional RNN with GRU units (BGRU) for the fine-scale f-RNN and a one-layer BGRU for the coarse-scale c-RNN. We set $d_f = 32$ and $d_c = 32$. For DCASE2017, we employ a single-layer BGRU as the f-RNN and a two-layer bidirectional RNN with LSTM units (BLSTM) as the c-RNN. We set $d_f = 16$ and $d_c = 64$. For both datasets, we set initial learning rate to 0.001 and optimize the binary cross entropy loss with Adam [16]. We use a fixed threshold of 0.5 on $\mathbf{y}^t$ in (2) and (3) to indicate the presence or absence of a sound event.

For both DCASE2016 and DCASE2017, we follow the official 4-fold cross validation partition and report the error rate (ER) on one-second segments [12] on all folds [13]. For each rotation, 3 folds form our training and validation sets with a ratio of 80:20. We train on the training set for 100 epochs and save the model that performs the best on the validation set. During testing, we ensemble the four best models from the four cross validation rotations by averaging their probability outputs. Finally, we binarize averaged output and

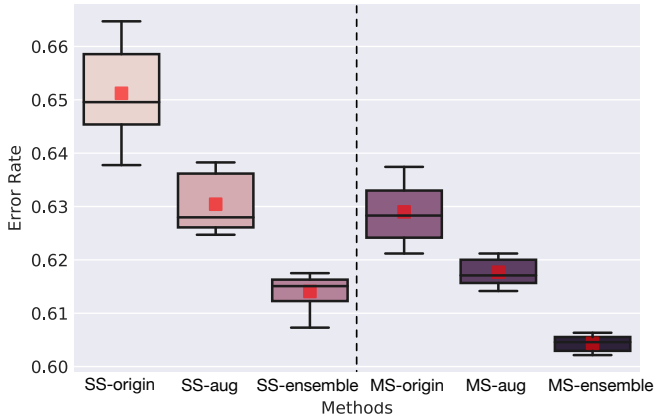

Fig. 2. Error rates of different approaches. "Origin": without data augmentation or ensemble; "Aug": with data augmentation; "Ensemble": with data augmentation and ensemble.

post-process it by median filtering with a kernel size of 27 frames [14]. We test the ER results for statistical significance using a paired student's t-test at a significance level of 0.05.

3.2. Single-Scale vs. Multi-Scale

To show the effectiveness of the proposed multi-scale model, we compare it with a state-of-the-art single-scale model that we previously proposed, which won the third place among the 13 teams in Task 3 of DCASE2017. The single-scale model is essentially the same as the fine-scale RNN of the proposed multi-scale model, on both its structure and training process. We use the best feature combination found in Section 2.2 for the single-scale RNN and the multi-scale RNN as input.

For the DCASE2017 development dataset, we run each experiment for 10 times and show error rates of different approaches as boxplots in Fig. 2. For both the single-scale (SS) and multi-scale (MS) models, we observe obvious reductions in error rates when data augmentation is applied. Moreover, ensemble decreases error rates to a new level and makes the predictions more stable (smaller variance). The final error rate of our SS-RNN ("SS-ensemble" in Fig. 2) is 0.614 ± 0.003, which exceeds the official baseline by over 11%. With the proposed MS-RNN, we achieve an error rate of 0.604 ± 0.001 ("MS-ensemble" in Fig. 2) which surpasses the baseline by over 12.5%. The reduction on ER is statistically significant with a p-value of $6.16 \times e^{-7}$ compared to the SS-RNN.

We exhibit error rates of the top 5 systems on both the evaluation set and development set of DCASE2017 in Table 1. Since the evaluation dataset of DCASE2017 is not publicly available, we cannot test the proposed MS-RNN on it. We believe that our model can achieve a lower ER on the evaluation dataset due to its stable performance. It is worth to mention that the top 2 systems [17, 18] achieve very low error rates on the development dataset, while their evaluation performances are quite similar to our SS-RNN model. The big gap between the evaluation and development set performances suggest severe overfitting on the development set, making it hard to compare with other systems on the development set.

Rank	Methods	Evaluation	Development
1	Conv RNN [17]	**0.791**	**0.25**
2	Multi-Input CNN [18]	0.808	0.51
3	Our SS-RNN [19]	0.825	0.614 ± 0.003
4	LSTM [20]	0.853	0.66
5	CNN [21]	0.858	0.81
9	DCASE2017 Baseline	0.936	0.69
-	Proposed MS-RNN	–	0.604 ± 0.001

Table 1. Error rates (mean with standard deviation) comparison of our proposed multi-scale (MS) RNN with the top 5 systems and the official baseline in Task3 of DCASE2017.

Methods	ER	F1 (%)
GMM [4]	0.91	23.7
FNN [14]	1.32 ± 0.06	$\mathbf{32.5 \pm 1.2}$
CRNN [6]	0.95 ± 0.02	30.3 ± 1.7
Our SS-RNN [19]	0.85 ± 0.01	32.1 ± 1.0
Proposed MS-RNN	$\mathbf{0.82 \pm 0.01}$	31.5 ± 0.8

Table 2. Comparison of our proposed multi-scale (MS) RNN with four other deep models on the development dataset of DCASE2016. All values are reported with mean and standard deviation.

For DCASE2016, we find that data augmentation leads to worse performance. We thus only employ ensemble for our final system. As shown in Table 2, we compare our SS-RNN baseline and the proposed MS-RNN with several other deep models on the development dataset of Task3 in DCASE2016. Compared to SS-RNN, MS-RNN shows a statistically significant reduction on ER with a p-value of $6.5 \times e^{-6}$. Moreover, MS-RNN outperforms other models by a large margin.

4. CONCLUSIONS

In this paper, we proposed a novel multi-scale RNN for sound event detection (SED) in real life. By integrating information from both time resolutions, we achieved lower error rates on Task3 in DCASE2016 and DCASE2017. The effectiveness of our proposed multi-scale model complies with the intuition that the combination of models from different time resolutions have the benefits of modeling both the fine-grained and long-term dependencies. Future directions include a combination of multiple time resolutions in the multi-scale framework to model even longer temporal dependencies. Another direction is to address data scarcity issue in SED by designing algorithms that can learn from weakly labeled datasets.

5. REFERENCES

[1] Daniele Barchiesi, Dimitrios Giannoulis, Dan Stowell, and Mark D Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[2] Toni Heittola, Annamaria Mesaros, Antti Eronen, and Tuomas Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, pp. 1, 2013.

[3] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *41th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[4] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference (EUSIPCO)*, Budapest, Hungary, 2016.

[5] Onur Dikmen and Annamaria Mesaros, "Sound event detection using non-negative dictionaries learned from annotated overlapping events," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013.

[6] Emre Cakır, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 6, pp. 1291–1303, 2017.

[7] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara, "Hierarchical boundary-aware neural encoder for video captioning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[8] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei, "A hierarchical approach for generating descriptive image paragraphs," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[9] Jongpil Lee and Juhan Nam, "Multi-level and multi-scale feature aggregation using pre-trained convolutional neural networks for music auto-tagging," *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1208 – 1212, 2017.

[10] Xinxing Li, Haishu Xianyu, Jiashen Tian, Wenxiao Chen, Fanhang Meng, Mingxing Xu, and Lianhong Cai, "A deep bidirectional long short-term memory based multi-scale approach for music dynamic emotion prediction," in *41th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[11] Justin Salamon and Juan Pablo Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[12] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, pp. 162, 2016.

[13] George Forman and Martin 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.

[14] Emre Cakir, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *International Joint Conference on Neural Networks (IJCNN)*, 2015.

[15] Sharath Adavanne, Pasi Pertilä, and Tuomas Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *42th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[16] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[17] Sharath Adavanne and Tuomas Virtanen, "A report on sound event detection with different binaural features," Tech. Rep., DCASE2017 Challenge, September 2017.

[18] Il-Young Jeong, Subin Lee, Yoonchang Han, and Kyogu Lee, "Audio event detection using multiple-input convolutional neural network," Tech. Rep., DCASE2017 Challenge, September 2017.

[19] Rui Lu and Zhiyao Duan, "Bidirectional GRU for sound event detection," Tech. Rep., DCASE2017 Challenge, September 2017.

[20] Jianchao Zhou, "Sound event detection in multichannel audio LSTM network," Tech. Rep., DCASE2017 Challenge, September 2017.

[21] Yukun Chen, Yichi Zhang, and Zhiyao Duan, "DCASE2017 sound event detection using convolutional neural network," Tech. Rep., DCASE2017 Challenge, September 2017.