# SOUND EVENT DETECTION IN REAL LIFE AUDIO USING MULTI-MODEL SYSTEM

*Yuanbo Hou, Shengchen Li*

Beijing University of Posts and Telecommunications
{hyb, shengchen.li}@bupt.edu.cn

## ABSTRACT

In this paper, we present a polyphonic sound event detection (SED) system based on a multi-model system. In the proposed multi-model system, we use one model based on Deep Neural Networks (DNN) to detect sound events of car, and five models based on Bi-directional Gated Recurrent Units Recurrent Neural Networks (BGRU-RNN) to detect other sound events including: brakes squeaking, children, large vehicle, people speaking and people walking. Since different classes sound events have different audio characteristics, we use different models to detect each class. The proposed multi-model system is trained and tested based on IEEE DCASE2017 Challenge: Sound Event Detection in Real Life Audio (Task 3) Development Dataset, the result yields up to 58.92% and 0.60 in terms of F-Score and error rate on segment-based metric respectively.

***Index Terms—*** Sound event detection, Multi-model

## 1. INTRODUCTION

Sounds carry a large amount of information about our everyday environment. Environmental sound analysis has attracted attention of many researchers recently. There are many examples of sound event detection (SED) applications, including audio classification, audio information retrieval and surveillance.

SED could be divided into monophonic detection and polyphonic detection. SED in single source conditions is called monophonic detection. SED in multiple sources conditions is called polyphonic detection. SED in real-life is usually a polyphonic detection as environmental sound is often a mixture audio that come from multiple sound sources simultaneously. Most polyphonic SED systems aim to recognize the beginning time, ending time and label of each sound event [1].

Traditional method for SED often use Mel Frequency Ceptral Coefficients (MFCC) [2] as features, then use classifier based on Gaussian Mixture Model (GMM) [3] or Hidden Markov Model (HMM) [4]. Most previous works focused on monophonic SED. In recent works, polyphonic SED is investigated by many researchers: Non-negative Matrix Factorization (NMF) [5][6] was used to analyze the number of sources in multiple sources. However, when the number of overlapping events is not known a priori, the fixed constraint of NMF on this number reduces its practicality. For overlapping events, a voting system [7] based on Generalized Hough transform (GHT) has been proposed to detect it. More recently, the emergence of methods using deep learning is noticeable. DNN has achieved good results on detecting overlapping sound events [8]. However, in DNN all observations are treated independently, it lacks context information. In order to keep context information, Bi-directional Long Short Term Memory (BLSTM) RNN [1] is proposed for polyphonic SED in real life recordings, which yields better result than DNN. The combination of Gated Recurrent Units [9], a GRU-RNN architecture is applied to SED of DCASE2016 challenge and achieved superior performance compared with the baseline.

IEEE DCASE2017 Challenge Task 3 is sound event detection in real-life audio, which evaluated the performance of sound event detection systems in multisource conditions. The number of predefined sound event classes were selected, and system detects the presence of these sounds. The selected sound classes in the task are: brakes squeaking, car, children, large vehicle, people speaking, and people walking [10].

The dataset [11] consists of recordings of street acoustic scenes with various levels of traffic and other activity. And in this task, there is no control over the number of overlapping sound events at each time, which means several sound events could happen simultaneously. The difficulties in this task are: 1) a certain number of classes are interested, other active sound events in street acoustic scenes may cause interference to the six selected sound event classes. For example, features calculated from multi-source audio may not match with the features that extracted from audio in isolation; 2) for each audio stream, the number of sound source is not clear; 3) several sound events may happen simultaneously; 4) different class of sound events have different characteristics, it is difficult to recognize six classes sound events with one single model.

Motivated by the good performance shown by the DNN in [12], and the flexibility in working with sequential data shown by RNN in [1], we propose to use multi-model system to detect multi-label multi-class sound event. In the paper, we obtain our experimental results by employing multi-model system (one model based on DNN and five models based on BGRU-RNN) to IEEE DCASE2017 Challenge Task 3. The remainder of this paper is organized as follows. Section 2 presents general information about DNN and RNN, and describes BGRU-RNN architectures. In Section 3, we propose our multi-model SED system. In Section 4, we conduct experiments of task 3, present our results and analysis it. Finally, Section 5 draws our conclusions.

## 2. RECURRENT NEURAL NETWORKS

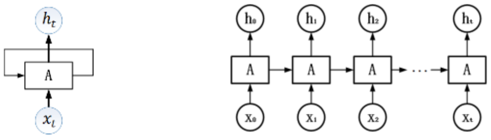
### 2.1. Deep Neural Networks

DNNs have been widely used in Acoustic Scene Classification (ASC), SED and audio tagging, which have shown good performance for these tasks [8]. DNN is a kind of feedforward neural network (FNN). Although DNN has shown good performance for SED, it may have difficulties processing sequential data, such as audio. Because in DNN all observations are treated independently of each other, it lacks context information.

The DNN we used in our experiment is a fully connected network with one hidden layer.

## 2.2. Recurrent Neural Networks

In order to keep past context information, feedback connections is proposed in neural network and this network architecture is called recurrent neural network (RNN). The feedback connections provide RNN with information circulate indefinitely, allowing information to persist (Fig.1a). In Fig.1a, A is a chunk of neural network, x is input and h is output. A loop allows information to be passed from one step of the network to the next. If we unroll the loop, RNN can be seen as multiple copies of the same network that is shown in Fig.1b, each passing a message to a successor.



(a) (b)

Fig.1: (a) Simple RNN, (b) Unroll the loop of RNN [18].

When information from a future timesteps are available, it can be used to provide future context information to the network using Bi-directional RNN (BRNN) [9]. In BRNN, second hidden layer learns input sequence in an inverse direction (Fig.2). The forward layer and backward layer could provide the network with context information that is full and symmetrical. Since the information to be predicted at each time step is from the backward and forward directions, better predictions should be made.

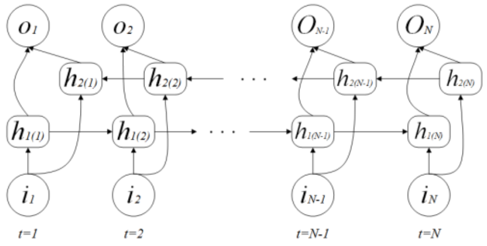
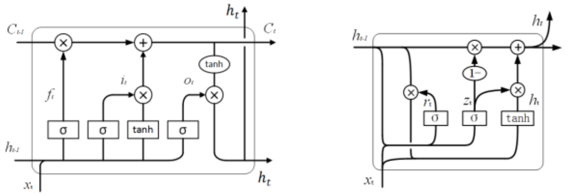


Figure 2: A BRNN architecture [6]

## 2.3. Bi-directional Gated Recurrent Units

In practice, simple RNN may be difficult to train and work efficiently, because of the exploding and vanishing gradient problem [11]. The problem makes RNN not able to deal with long-range dependencies. In order to solve this problem, two variants of RNNs are proposed: Long Short Term Memory networks (LSTMs) [12] and Gated Recurrent Units (GRUs) [13]. In LSTM-RNN, the neurons of simple RNN are substituted by LSTM blocks (Fig. 3a). LSTM block contains three gating neurons: input, forget and output, and all gating neurons choose different information by the logistic function. Compared with simple RNN, LSRM-RNN does not have exploding and vanishing gradient problem. In GRU-RNN, the neurons of simple RNN are substituted by GRU blocks (Fig. 3b). Different with LSTM block, GRU block combines the 'input gate' and 'forget gate' of LSTM into 'update gate', merges cell state and hidden state.



(a) (b)

Fig.3: (a) LSTM block, (b) GRU block [18].

GRU block is simpler than LSTM block and its' computational cost is lower than LSTM [9]. In sequence modeling, GRU has shown comparable performance to LSTM [13]. Hence, BGRU-RNN is proposed in this task, because Bi-directional RNN (BRNN) could take advantage of both past context information and future context information. It is supposed to get better prediction. In our multi-model system, a BGRU-RNN is obtained by substituting the neurons of simple RNN by GRU units.

## 3. THE PROPOSED MULTI-MODEL SED SYSTEM

The selected sound classes for task 3 are: brakes squeaking, car, children, large vehicle, people speaking, and people walking. The recordings come from street acoustic scenes with various levels of traffic and other activity, and different class of sound events have different audio characteristics, there are obvious differences between them. For each class, the statistics of event instances in development dataset are shown in the Table 1.

In baseline system, DNN has good performance for detecting sound events of car compared with other five classes sound events. And from Table 1, we know that the number of car instances is the largest. The sound events of car have longer duration and bigger energy than other class sound events. Motivated by the good performance shown by DNN, we propose to use DNN-based model to detect "car" events and other five models based on BGRU-RNN to detect other sound events respectively.

### 3.1. Multi-Model System

#### *3.1.1. Car Model*

Car model based on DNN only to detect whether there are audio events of "car" or not in per frame. For car model, the system consists of two stages. First, log mel-band energies are extracted from raw audio data as features. Then, features are used for training the classifier. Second, sound events are detected by model and smoothing the output of the model.

First stage is extracting features, the input of car model are raw audio data, they have different recording conditions. To account for this problem, the magnitude of each recording is normalized to [-1, 1]. The audio stream are split into 40 millisecond (ms) frames with a 50% overlap. We calculate the log amplitude of each frame within 40 mel-bands and normalize each frequency band [1].

Table 1: The statistics of per class in development dataset

| *Event class* | *Brakes squeaking* | *Car* | *Children* | *Large vehicle* | *People speaking* | *People walking* |
|---|---|---|---|---|---|---|
| *Number* | 52 | 304 | 44 | 61 | 89 | 109 |
| *Shortest duration* | 0.344 s | 0.510 s | 0.533 s | 0.383 s | 0.285 s | 0.296 s |
| *Average duration* | 1.865 s | 8.156 s | 7.977 s | 15.143 s | 8.041 s | 11.440 s |
| *Longest duration* | 12.478 s | 86.032 s | 202.529 s | 85.544 s | 137.676 s | 106.853 s |
| *Median of duration* | 1.177 s | 6.002 s | 1.988 s | 12.000 s | 3.910 s | 4.910 s |
| *Standard deviation of duration* | 2.009 s | 7.543 s | 30.320 s | 13.113 s | 15.928 s | 17.893 s |

Second stage is training the model. Car model consists of one fully connected layer of 40 hidden units with 20% dropout, the output layer of it contains two units and activation of hidden layer is Rectified Linear Unit (ReLU). The network is trained using Adam algorithm for gradient-based optimization [10]. Other information of car model is shown in Table 2.

### *3.1.2. Brakes Squeaking Model*

Compared with "car" sound events, "brakes squeaking" (BS) is not easy to detect because it has less samples, its average duration is shorter than other classes sound events, and its sound is relatively small. For BS, we propose to use BS model based on BGRU-RNN to detect it. BS model for "brakes squeaking" only to detect whether there are audio events of "brakes squeaking" or not in per frame.

In model based on BGRU-RNN, we are not using the feature extracted from audio data, but directly using the raw audio data. The training data are split in frame of 20 ms, as we know, the sampling rate of the raw audio is 44.1 KHz. So, for per frame of audio, we get 882 samples. Then, we divided the 882 samples of each frame into 9 copies, meaning that, each copy containing 98 original samples. At each time step, a copy of the samples is sent into the network for training. BS model consists of one hidden layer and one output layer, the hidden layer is a BGRU-RNN layer, information of BS model is shown in Table 2.

### *3.1.3. Large Vehicle Model*

Due to the fewer instances than other class, "large vehicle" (LV) is difficult to detect in street acoustic scenes with various levels of traffic and many other activities. For LV, we propose to use LV model based on BGRU-RNN to detect it. The configuration of LV model is shown in Table 2.

### *3.1.4 People Speaking Model*

Recordings come from street acoustic scenes with various levels of traffic and many other activities. In audio, the voice of people speaking (PS) is easily disturbed by noise, which makes it difficult to detect people speaking. For PS, we propose to use PS model based on BGRU-RNN to detect it. The configuration of PS model is shown in Table 2.

### *3.1.5 People Walking Model*

The recordings contain other sounds similar to footsteps, making the footsteps are hard to detect. For "people walking" (PW), we propose to use PW model based on BGRU-RNN to detect it.

PW model for "people walking" different with car model, in PW model output layer containing 6 units. Other configuration of PW model is shown in Table 2.

### *3.1.6 Children Model*

From Table 1, we know that "children" has the longest duration and the largest standard deviation of duration compared with other class events. And in baseline system based on DNN, the recognition result of "children" class event has the highest error rate. For "children", we propose to use children model based on BGRU-RNN to detect it.

In children model, the stream are split into 20 ms frames, and frames were constructed using a 9-frame context, resulting in a vector length of 8982. At each time step, a frame samples is sent into the network for training. Information of children model is shown in Table 2. The first hidden layer is a BGRU-RNN layer, and next two hidden layers are fully connected layers.

Table 2: Configuration of Multi-Model

<table>
<tr><th><i>Model</i></th><th><i>Car</i></th><th><i>BS</i></th><th><i>LV</i></th><th><i>PS</i></th><th><i>PW</i></th><th><i>Children</i></th></tr>
<tr><td><i>Based</i></td><td>DNN</td><td colspan="5">BGRU-RNN</td></tr>
<tr><td><i>Timesteps</i></td><td>Null</td><td colspan="5">9</td></tr>
<tr><td><i>Input</i></td><td>Mel features</td><td colspan="5">Raw audio data</td></tr>
<tr><td><i>Units of input layer</i></td><td>40</td><td colspan="4">98</td><td>882</td></tr>
<tr><td><i>Number of hidden layer</i></td><td>1</td><td colspan="4">1</td><td>3</td></tr>
<tr><td><i>Units of hidden layer</i></td><td>40</td><td colspan="4">50</td><td>(50, 100, 100)</td></tr>
<tr><td><i>Activation of hidden layer</i></td><td colspan="6">ReLU</td></tr>
<tr><td><i>Units of output layer</i></td><td colspan="4">2</td><td colspan="2">6</td></tr>
<tr><td><i>Activation of output layer</i></td><td colspan="6">Sigmoid</td></tr>
</table>

### 3.2. Post-processing

#### *3.2.1. Smooth output*

In real life, the audio stream is always continuous, which means the audio stream is highly impossible to change the audio class too suddenly or too frequently. Under this assumption, we propose to use smoothing progress in the output label of an audio sequence. Assume the output label sequence is $s_1, ..., s_n$, then for i < n we do smoothing as:

$$\begin{aligned} & if \quad s_{i-1} \neq s_i \;\&\&\; s_{i-1} = s_{i+1} \\ & then \quad s_i = s_{i-1} \end{aligned}$$

The rule [14] implies that if the middle index label is different from the other two label, while the other two are the same, then, the middle label is considered as misclassification.

#### *3.2.2. Delete too short predicted events*

In real life, audio stream is always continuous and sound event will last for a while. In experiment, we find that too many short duration outputs would lead to an increase in error of insertion rate. To reduce insertion rate error, for per class, first, we calculate the duration of each instances, removing the maximum and minimum values. Then, we take the shortest duration as our threshold. If the duration of recognized sound event is less than the threshold, we think it did not occur and delete it. Threshold are shown in Table 3.

Table 3: Threshold of per class

| *Sound event* | *Threshold* |
|---|---|
| *Brakes squeaking* | 0.36 s |
| *Car* | 0.60 s |
| *Children* | 0.60 s |
| *Large vehicle* | 1.25 s |
| *People speaking* | 0.36 s |
| *People walking* | 0.38 s |

After the two stages of post-processing, we extract the results of corresponding sound events from each model, to form our final results.

## 4. EXPERIMENTAL RESULTS AND ANALYSIS

In this task, the results are evaluated using segment-based error rate and segment based F-score as metrics [10], using a segment length of one second. The four cross-validation folds are treated as a single experiment: the metrics are calculated by accumulating error counts over all folds, not by averaging the individual folds nor the individual class performance. This method [15] of calculating performance gives equal weight to each individual sound instance in each segment, as opposed to being influenced by class balance and error types. Our multi-model system was trained and tested based on Development Dataset, using the provided cross-validation setup obtained an overall error rate of 0.60 and an overall F-score of 58.92%, as shown in Table 4. For completeness, individual class performance is presented along the overall performance.

As shown in Table 4, our multi-model system achieves better recognition than baseline system in some classes. In the experiment, we find that for sound event detection, too many hidden layers of networks do not achieve better recognition results. In car model, our DNN contains only one hidden layer, and achieves the performance comparable to baseline system. If we add a fewer hidden layers on car model, the recognition of “car” in model will be worse, not better. For simple model with fewer hidden layers, we find that the main error is the deletion rate. For complex model with more hidden layers, we find that the main error is the insertion rate. This means that for many sound events in development dataset, simple models do not recognize them, and complex models may be a bit over-fitting.

In real-life audio, different class sound events have different audio characteristics, a single model is difficult to effectively recognize all class sound events and achieves good performance on every class sound events. For example, children model is a multi-class classifier, compared to baseline system, in children model, the recognition of “children” yields up to 0.66, approximately 70% higher than the baseline. In children model, the recognition for “children” gets good performance, but for other class events is relatively poor.

Considering the difficulties in sound event detection in real life, for polyphonic sound event detection, we propose to use multi-model system to detect multiple classes sound events. Based on task 3 development dataset, we implement the proposed multi-model system and get better result than baseline system.

Table 4: Results for Task 3, segment-based metrics

| *Overall* | *Multi-Model System* | | *Baseline* | |
|---|---|---|---|---|
| | *ER* | *F-Score* | *ER* | *F-Score* |
| | 0.60 | 58.92% | 0.69 | 56.7 % |
| *Brakes squeaking* | 1.0 | NaN | 0.98 | 4.1% |
| *Car* | 0.56 | 72.1% | 0.57 | 74.1% |
| *Children* | 0.66 | 51.4% | 1.35 | 0.0% |
| *Large vehicle* | 0.76 | 55.1% | 0.90 | 50.8% |
| *People Speaking* | 1.03 | 1.0% | 1.25 | 18.5% |
| *People Walking* | 0.71 | 60.2% | 0.84 | 55.6% |

## 5. CONCLUSION

In this paper, we proposed to use multi-model system for polyphonic sound event detection in real life. Based on DNN and BGRU-RNN, the proposed approach outperforms the baseline system tested on development dataset and obtains an overall error rate of 0.60 and an overall F-score of 58.92%. For multi-label polyphonic sound events detection, multi-model system showed their effectiveness and flexibility compared to baseline system.

Future work will concentrate on detecting the sound events of “people speaking” in real life, because compared with other class sound events, the recognition of “people speaking” is relatively poor.

## 6. REFERENCES

[1] T. V. Giambattista Parascandolo, Heikki Huttunen, "Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings," no. May, 2016.

[2] J. Kaur, "Environment Independent Speech Recognition System using MFCC ( Mel-frequency cepstral coefficient )," vol. 6, no. 5, pp. 4186–4190, 2015.

[3] L. Vuegen, B. Van Den Broeck, P. Karsmakers, J. F. Gemmeke, and B. Vanrumste, "An MFCC-GMM approach for event detection and classification," *IEEE AASP Chall. Detect. Classif. Acoust. Scenes Events*, no. 2, pp. 2–4, 2013.

[4] X. Zhou, X. Zhuang, M. Liu, H. Tang, M. Hasegawa-Johnson, and T. Huang, "HMM-based acoustic event detection with adaboost feature selection," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4625 LNCS, pp. 345–353, 2008.

[5] J. F. Gemmeke *et al.*, "AN EXEMPLAR-BASED NMF APPROACH TO AUDIO EVENT DETECTION ESAT-PSI , KU Leuven , Kasteelpark Arenberg 10 , 3001 , Leuven , Belgium," no. 1, pp. 3–6, 2013.

[6] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2015–Augus, pp. 151–155, 2015.

[7] J. Dennis, H. D. Tran, and E. S. Chng, "Overlapping Sound Event Recognition using Local Spectrogram Features with the GeneGeneral Hough Transform," *Proc. Annu. Conf. Int. Speech Commun. Assos.*, pp. 2266–2269, 2012.

[8] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "28-Polyphonic sound event detection using multi label deep neural networks," in *2015 international joint conference on neural networks ({IJCNN})*, 2015, pp. 1–7.

[9] J.-C. W. Toan H. Vu, "ACOUSTIC SCENE AND EVENT RECOGNITION USING RECURRENT NEURAL NETWORKS Toan H . Vu , Jia-Ching Wang National Central University Department of Computer Science and Information Engineering Taoyuan , Taiwan," no. September, pp. 3–5, 2016.

[10] A. Mesaros *et al.*, "DCASE 2017 CHALLENGE SETUP : TASKS , DATASETS AND BASELINE SYSTEM Tampere University of Technology , Laboratory of Signal Processing , Tampere , Finland Carnegie Mellon University , Department of Electrical and Computer Engineering , & Department of Langu," no. November, 2017.

[11] A. Mesaros, T. Heittola, and T. Virtanen, "TUT Database for Acoustic Scene Classification and Sound Event Detection."

[12] Q. Kong, I. Sobieraj, W. Wang, and M. Plumbley, "Deep Neural Network Baseline for Dcase Challenge 2016," *Dcase*, no. September, pp. 4–8, 2016.

[13] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training Recurrent Neural Networks."

[14] S. Schmidhuber, Jürgen, Hochreiter, "LONG SHORT-TERM MEMORY," vol. 9, no. 8, pp. 1–32, 1997.

[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv*, pp. 1–9, 2014.

[16] L. Peng, D. Yang, and X. Chen, "Multi frame size feature extraction for acoustic event detection," *2014 Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. APSIPA 2014*, 2014.

[17] G. Forman, M. Scholz, G. Forman, and M. Scholz, "Apples-to-Apples in Cross-Validation Studies : Pitfalls in Classifier Performance Measurement Abstract : Apples-to-Apples in Cross-Validation Studies : Pitfalls in Classifier Performance Measurement," vol. 12, no. 1, 2009.

[18] http://colah.github.io/posts/2015-08-Understanding-LSTMs/