DCASE2017 SOUND EVENT DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

Yukun Chen, Yichi Zhang and Zhiyao Duan

Department of Electrical and Computer Engineering, University of Rochester, USA

ABSTRACT

The DCASE2017 Challenge Task 3 is to develop a sound event detection system of real life audio. In our setup, we merge the two channels into one, then use Mel-band energy to calculate the converted spectrum, and train the model using a convolutional neural network (CNN). The method we use achieves a 0.8575 segment-based error rate on the final result, which is an 8.4% improvement comparing to the baseline model. It proves the practicability of using CNNs for sound event detection.

Index Terms— Sound event detection, Mel-filter bank, Convolutional neural network, Acoustic scene classification

1. INTRODUCTION

Sound event detection (SED) is a relatively new research area. Researchers have been designing various systems to detect the scenes and events in provided audio files. SED are useful in but not limited to the following fields: smart home electrical appliance development [1], video classification [2], and security [3].

The DCASE2017 Challenge Task 3 uses training and testing materials recorded in real-life environments [4]. Participants are required to label the correct time frames for specific events in six categories. One of the difficulties is the unlimited number of overlapping sound events at each time. Other difficulties include the unbalanced categories, varying length of events, etc.

Typical AED frameworks are composed of at least two parts: feature extraction and audio event inference [5]. In our method, we use Mel-band energy as our feature extraction process and convolutional neural network (CNN) as the inference process. CNN is widely used in image processing [6] as well as speech recognition [7]. It is chosen for the ability to learn internal characteristics from the large training dataset.

2. SYSTEM SETUP

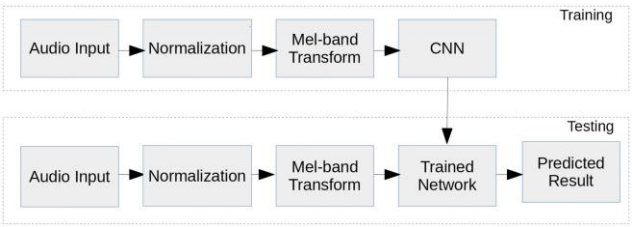

Figure 1: Framework of the proposed system.

As shown in Figure 1, the system we develop mainly consists of four different parts: normalization, feature extraction, convolutional neural network (CNN) and post-processing. We first use training audio to train and tune the model, then predict on the testing audio according to the trained model. Details of each part are discussed below:

2.1. Normalization

First, we mix the two channels into one by calculating their average value. For the audio files provided, one of the problems is the inconsistency of sound volume. To deal with this, we normalize all the audio files per their maximum amplitude. After normalization, all the audio files roughly maintain at the same volume in order to better fit the training model.

2.2. Feature Extraction

We calculate Mel-band energy spectrum in our feature extraction step. Mel-band energy spectrum is widely used in sound event detection (SED) [8] and source classification [9]. It transforms audio waveform into spectrum, and sooner we will deal with it in a way similar to image processing in the next step.

For detail parameters, our input audio has a sampling rate of 44.1 kHz. We use the LIBROSA package [ref] to calculate the Mel-band energy, set-ting the length of FFT window to 2048 points, hop size to 512 points, number of Mel bands to 64 and maximum frequency to 6400 Hz. Then, to get the training dataset, we merge every 128 frames (1,486ms) together into one patch. The reason we set this length is because that one important category, walking footsteps, has an impulse frequency of 1~2 Hz, the length is designed to cover at least 2 impulses. Up till this step, we have converted the audio wav files into a series of 64*128 arrays.

2.3. Convolutional Neural Network (CNN)

The deep neural network (DNN) outperforms most of other models (SVM, Random Forest, etc.) in complicated pattern recognition tasks [10]. DNN is able to model complex nonlinear relationships between inputs and outputs. Most DNNs, except the Recurrent Neural Networks (RNN), has a feedforward structure, where data flows from the input layer to the output layer without looping back [11].

Within DNN models, CNNs are widely used in computer vision and acoustic modeling. It consists of an input and an output layer, as well as multiple hidden layers. The hidden layers

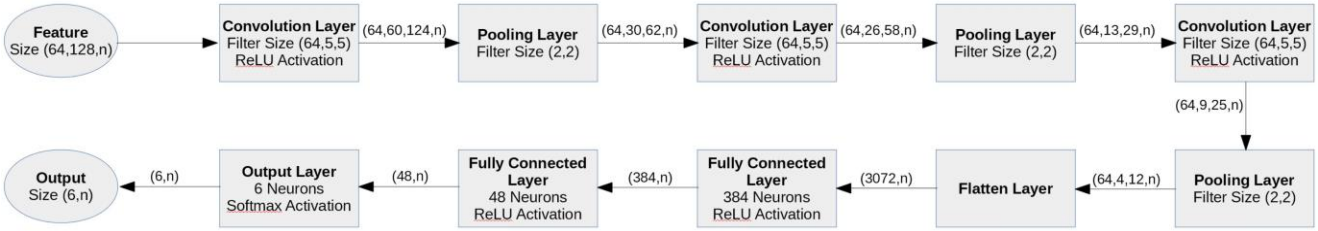

Figure 2: Structure of proposed CNN.

are either convolutional, pooling or fully connected [12], where convolutional layers emulate the response of an individual neuron to visual stimuli; pooling layers de-crease the complexity of features; and fully connected layers connect every neuron in one layer to every neuron in another layer, sharing same principle as the traditional multi-layer perception neural network (MLP).

In our configuration of CNN (shown as Figure 2), we use three sets of convolutional and pooling layers, with each set containing a convolutional layer of size (64, 5, 5) and a maximum pooling layer of size (2, 2). Passing through each set of layers, the complexity of features declines gradually, and finally turns to an array of size (12, 4, 64). We then flatten the array to one dimension with size 3,072, and pass it to the fully connected layers with two hidden layers using ReLU activation. The sizes of hidden layers are 384 and 48, and the final output layer has a size of 6, with values of the neurons ranging from 0 to 1 indicating the likelihood of the presence of the corresponding sound events. The whole network takes input features of size (64, 128, n) and transfers it to an array of size (6, *n*), where *n* is the number of patches of either the training or the test set.

2.4. Post-processing

The post-processing step translates the 6**n* array output into the required format containing the start and end time of each sound event. Since the output value of each neuron is set between 0 and 1, we need to set a threshold value to determine the presence of the event. The value we finally choose is 0.4 since we find out that the total number of detected events is significantly smaller than that of correctly labeled events in our testing. Comparing to the 0.5 threshold, setting it to 0.4 can increase the performance by around 4%. Besides, we also apply median filter to reduce the outliers, and fuse the prediction results of evaluation set from the four cross-validation models together.

3. EXPERIMENTAL RESULTS

Table 1: Overall result of our developed system compared to the baseline system and the best system [13]

Segment-Based Result	**Overall / Evaluation dataset**		**Overall / Development dataset**	
	Error Rate	F-measure	Error Rate	F-measure
Our System	0.8575	30.9%	0.81	37.0%
Baseline System	0.9358	42.8%	0.69	56.7%
Best System	0.7914	41.7%	0.25	79.3%

The DCASE 2017 Task 3 result is based on the sound event detection accuracy of given audio tracks with labeled events.

From Table 1, we can see that our system outperforms the baseline system in terms of the error rate on the evaluation dataset. On the other hand, the error rate of the baseline system on the development dataset is better than ours. This difference may suggest that the baseline system overfits the development dataset. However, by comparing to the system with the best result in this competition we find that the result of their development dataset is even better. Hence, there shall be no direct relation between the result of evaluation and development dataset. Similar conclusion can be drawn for the error rate and F-measure of evaluation dataset as well. A better error rate cannot ensure better performance in F-measure.

Table 2: Class-wise result of our developed system compared to the baseline system and the best system.

	Our System		**Baseline System**		**Best System**	
	Error Rate	F-measure	Error Rate	F-measure	Error Rate	F-measure
Brakes Squeaking	1.000		0.921	16.5%	1.000	
Car	0.854	51.8%	0.767	61.5%	0.767	54.6%
Children	1.000		2.667	0.0%	1.200	0.0%
Large Vehicle	0.989	14.6%	1.441	42.7%	1.068	49.3%
People Speaking	1.008	0.0%	1.298	8.6%	1.041	0.0%
People Walking	1.066	1.0%	1.445	33.5%	1.033	38.7%

We try to look at the detail of difference according to the class-wise comparison. From Table 2, we find that for all three systems, their prediction of the *car* category is the most accurate among all categories. For other categories, the error rate of our system is similar to the best system, but the F-measure is very different in some categories. The prediction by the baseline system, however, is not so good in categories other than *car*. We investigate the result and try to find out possible explanations. The reason that all three models predict relatively well in *car* category may be due to that the segment length of car category dominates the development dataset, so all models manage to learn the feature relatively well. For categories other than *car*, the baseline system tends to give large amount of incorrect prediction, which may be cause by overfitting issue. Our system, in comparison, gives little amount of prediction, which is mostly incorrect as well. Reasons for this may be due to that we set a very slow learning rate for CNN training, and use early stopping technique to prevent overfitting. Thus, the system tends to be

conservative and makes little prediction for uncertain categories. The system that has the best performance, like the baseline system, gives a reasonable amount of correct predictions for categories other than *car*. However, its accuracy is much higher than the baseline system.

4. CONCLUSION

From our results, we can see that the CNN structure is able to effectively label most of the detected events in the correct time range. Although CNN structure performs relatively well at recognizing complex patterns, in DCASE2017 SED dataset, it seems that our CNN system can only capture limited regular patterns among features in the same class. The difference of patterns among various classes may not be sufficient for the system to distinguish.

To further improve the system, future work can be done by 1) training the model on additional datasets to summarize more general patterns and 2) pre-training the model with clearly labseled sound without background noise.

5. ACKNOWLEDGEMENT

This work is funded by the National Science Foundation grant No. 1617107. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

6. REFERENCES

[1] A. Harma, M. F. McKinney, and J. Skowronek, "Automatic surveillance of the acoustic activity in our living environment," in Proc. *IEEE International Conference on Multimedia and Expo (ICME)*, 2005.

[2] J. Portelo, M. Bugalho, I. Trancoso, J. Neto, A. Abad, and A, Serralheiro, "Non-speech audio event detection," in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.

[3] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in Proc. *IEEE Conference on Advanced Video and Signal Based Surveillance*, 2007.

[4] http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/

[5] D. Sadlier, and N. Oconnor, "Event detection in field sports video using audio-visual features and a support vector Machine," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1225-1233, 2005.

[6] P. Simard, D. Steinkraus, and J. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in Proc. *Seventh International Conference on Document Analysis and Recognition*, 2003.

[7] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition" in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[8] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[9] L. Lu, S. Z. Li, and H. Zhang, "Content-based audio segmentation using support vector machines," in Proc. *IEEE International Conference on Multimedia and Expo (ICME)*, 2001.

[10] S. Adavanne, P. Pertila, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[11] https://en.wikipedia.org/wiki/Deep_learning

[12] https://en.wikipedia.org/wiki/Convolutional_neural_network

[13] http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-sound-event-detection-in-real-life-audio-results