DEEP LEARNING FOR DCASE2017 CHALLENGE

An Dang, Toan H. Vu, Jia-Ching Wang

Department of Computer Science and Information Engineering, National Central University, Taiwan
andtt.cit@gmail.com, toanvuhong@gmail.com, jcw@csie.ncu.edu.tw

ABSTRACT

This paper reports our results on all tasks of DCASE challenge 2017 which are acoustic scene classification, detection of rare sound events, sound event detection in real life audio, and large-scale weakly supervised sound event detection for smart cars. Our proposed methods are developed based on two favorite neural networks which are convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Experiments show that our proposed methods outperform the baseline.

Index Terms— CNN, RNN, DenseNet, acoustic scene classification, sound event detection

1. INTRODUCTION

Recently, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been applied much for various audio recognition such as music tagging [1], acoustic scene classifi-cation [2, 3], and sound event detections [4, 5, 6]. CNNs provide an effective way to capture spatial information of multidimensional data, while recurrent neural networks (RNNs) are powerful in learning temporal sequential data. In this paper, we proposed various deep learning models based on CNNs and RNNs to perform the acoustic scene classification and sound event detection of DCASE 2017 challenge. The remainder of paper is organized as follows. In section 2, we present about proposed models. The experimental results for all tasks are discussed in section 3. Finally, conclusions are given in section 4.

2. PROPOSED METHODS

2.1. Task 1: Acoustic scene classification

In this task we proposed three different approaches that describe as follow. For the first approach we use mel frequency cepstral coefficients (MFCC) as features and convolutional recurrent neural networks (CRNN) as classifier method. We extract 60 MFCC features which are 20 first coefficients, 20 coefficients of the second derivative with three different window sizes of 0.02, 0.04 and 0.06 seconds and a hop size of 0.02 seconds. We, then, concatenate three 60 MFCC along frequency axis to form 180 dimensional features. These features are fed into a CRNN model. We design a CRNN architecture including two sequential CNN blocks and one GRU layer on top before a fully-connected layer. In particular, each sequential block consists of a convolutional layer with kernel size of 3, a ReLU activation function, and a batch normalization [7]. The output of each CNN block is concatenated with the previous input along frequency domain to increase the number of features before performing a max pooling. After each max pooling, we adopt a dropout [8] of 50% to reduce over-fitting problem. The output of last pooling layer are fed into a GRU layer which is suitable for modeling the data across long time scales, and finally, a fully connected is produced before predicting the class of the inputs.

For the second approach, the used features are 40 log-mel filter banks which are extracted from frames with size of 0.04 and hop size of 0.02 seconds. We employ them as input features to the CNN model. The CNN model is designed as the same as the CNN part of the first approach with a little different on top of the network. We use two fully-connected layers on top before a sofmax layer.

For the third approach, features are constructed from mel frequency cepstral coefficients (MFCC) and log-mel filter banks. We apply multi-scale windows for both features. In particular, we utilize three different window sizes of 0.02, 0.04, and 0.06 seconds, and a hop size of 0.02 seconds. For each pair of window size and hop size as a scale, we extract 20 MFCC features with their first and second temporal derivatives, which forms 60 dimensional features and 40 log-mel filter bank coefficients. After that, we concatenate these features along frequency axis, which results in 300-dimensional features. We feed these features into a CNN model as similar as the model of the second approach.

The all input features are normalized by subtracting mean and dividing them with standard deviation computed over the training set. In training process, we segment audio signal to short samples with length of 2 seconds which is time length of 100. We train network using the cross-entropy objective function. Adam [9] with a learning rate of 0.001 is employed as optimization method.

2.2. Task 2: Detection of rare sound events

This task concentrates on detection three independent events including baby crying, glass breaking, and gunshot. We use 40 log mel filter banks with 40 ms window size and 20 ms hop size for baby crying and glass breaking events. The used features for gun-shot event are 60 MFCC features, including the first 20 static coefficients, 20 delta coefficients, and 20 acceleration coefficients using 0.04 seconds window size and 0.02 seconds hop size. These features are fed into a pCRNN model as illustrated in Fig.2.

We develop a parallel convolutional neural network and recurrent neural network that is referred as pCRNN. Both CNN and RNN blocks also receive a context window of log-mel band ener-gies as their inputs. The CNN network is composed of ten convolutional layers with ReLU activation function, and a max-pooling layer after each two consecutive convolutional layers. The RNN network include a bidirectional gated recurrent unit layer (BiGRU) and a single gated recurrent unit layer (GRU). The outputs of two networks are merged and fed into a fully connected layer. We adopt a dropout of 50% after each pooling layer. We use binary cross entropy loss and Adam optimizer with learning rate of 0.001 for training network.

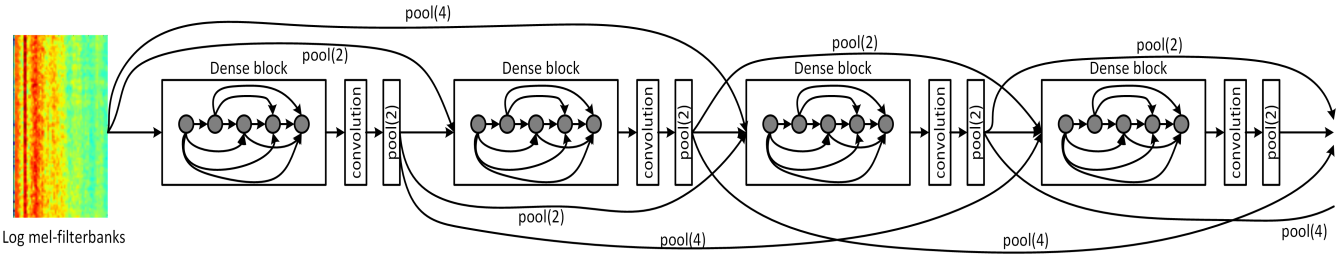

Figure 1: Our DenseNet model.

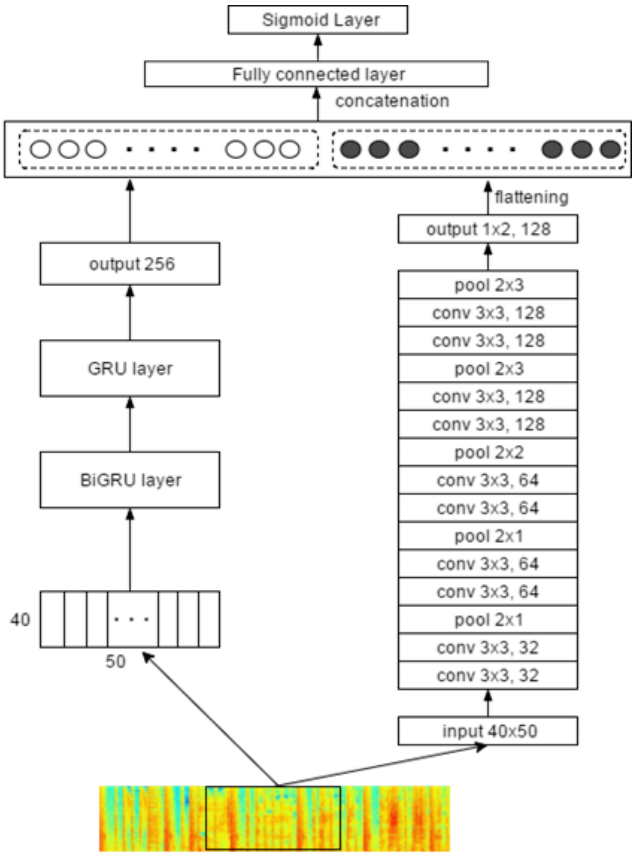

Figure 2: Parallel Convolutional Recurrent Neural Networks for Task 2&3

2.3. Task 3: Sound event detection in real life audio

Recordings are converted into time-frequency representation using 20 ms window size with Hamming window and 10 ms hop size. The input for the system is 60 MFCC features, including the first 20 static coefficients, 20 delta coeffi-cients, and 20 acceleration coeffi-cients. The features are normalized to zero mean and unit variance. We segment each audio feature into a short window of 1s and feed into pCRNN network.

Table 1: The acoustic scene classification performance

Model	Features	Accuracy
Baseline [11]	MFCC	74.8
CNN	Log-mel	79.1
CNN	MFCC + Log-mel (Multiscale)	81.6
CRNN	MFCC (Multiscale)	**82**

2.4. Task 4: Large-scale weakly supervised sound event detection for smart cars

Audios are resampled to 22050 Hz. We extract 64 log mel-filter banks with window size of 1024 and hop size of 256 to use as input features. Given an input, we segment it to regions. Each region goes through a classifier with *sigmoid* outputs. In this report, we introduce a simple method to combine these outputs to get the final output of the entire input sequence for the classification task. The final output o_e of an event e is derived as

$$o_e = \sum_i o_e^i att^i \tag{1}$$

where o_e^i is the output value of event e w.r.t the i^{th} region, att^i is an attentional weight that control how much a region contribute to final decision of the classification task. The attentional weight can be computed in various way. However, due to the limitation of time, we simply compute att^i as

$$att^i = \frac{o_e^i}{\sum_j o_e^j} \tag{2}$$

After training on the classification task, we can use the classifier as a detector to recognition sound events through time. Notice that we only employ the weak labels of both training set and development set in the training process. The optimal set of parameters is determined at the peak of F-score of the development set. Moreover, we develop a DenseNet [10] model as a classifier for the task. Figure 1 illustrates our model. The model consists of six dense blocks, each block has 4 layers with a growth rate of 20. To make the model denser, input of a dense block is comprised of not only the output of its previous block, but also from outputs of two blocks before the previous one.

3. EXPERIMENTAL RESULTS

In this part, we discuss the performance of the proposed methods and compare to the DCASE 2017 challenge baseline system [11].

Table 2: Class-wise accuracy of the baseline system and CRNN model for Task 1.

Scene	Baseline [11]	CRNN
beach	75.3	85.6
bus	71.8	90.4
cafe/restaurant	57.7	66.0
car	97.1	98.4
city center	90.7	89.1
forest path	79.5	92.3
grocery store	58.7	89.4
home	68.6	72.3
library	57.1	62.5
metro station	91.7	86.9
office	99.7	99.0
park	70.2	69.9
residential area	64.1	78.5
train	58.0	59.9
tram	81.7	89.4
Average	74.8	82.0

3.1. Task 1: Acoustic scene classification

In table 1, we compare the performance of the baseline system to our proposed methods on TUT Acoustic scene DCASE 2017 challenge development set. Our proposed method outperform the baseline system. The CRNN model using MFCC features with multi-scale window size obtain better performance than other features, improving 7.2% in comparison with the baseline.

Additionally, table 2 presents a class-wise accuracy of the baseline system and the best our model CRNN. Although, the results of some scenes such as *city-center*, *park*, and *metro station* exhibited a little bit worse accuracy than the baseline. However, several scenes such as *grocery store* and *bus* achieve a significant increment in accuracy from 58.7, 71.8 to 89.1, and 90.4, respectively.

3.2. Task 2: Detection of rare sound events

Table 3 shows the results in event-based error rate and F-score of baseline system and our pCRNN model for baby cry, glass break, and gunshot events. Our proposed model improve significantly the performance with a large margin of 28% in error rate and 13.7% in f-score in comparison to the baseline system.

3.3. Task 3: Sound event detection in real life audio

In table 4, we compare the average score in both terms error rate and f-score of model baseline to our pCRNN model. The pCRNN model improve the error rate performance from 0.69% to 0.59% in comparison with baseline. Though, the performance on F-score show slightly lower accuracy.

Table 3: Results in event-based error rate (ER) and F-score of our pCRNN model and baseline system for Task2

Event	Error rate		F-score	
	Baseline [11]	pCRNN	Baseline [11]	pCRNN
Baby cry	0.67	0.22	72.0	88.5
Glass break	0.22	0.16	88.5	91.6
Gunshot	0.69	0.37	57.4	79.2
Average	0.53	**0.25**	72.7	**86.4**

Table 4: Results in segment-based ER and F-score for task 3

Model	Error rate	F-score
Baseline [11]	0.69	56.7
pCRNN	0.59	55.4

3.4. Task 4: Large-scale weakly supervised sound event detection for smart cars

Besides parameters of models, our system relies much on several hyperparameters, such as thresholds for outputs, size of median filter for postprocessing. Particularly, optimal thresholds for outputs of systems are determined by a grid-search in the training processing. Each class has its own threshold that maximizes F-score of that class on the development set for the classification task. Table 5 and 6 present our best results on the two subtasks: audio tagging and sound event detection. These tables show that our results outperform the baseline system by a large margin.

Table 5: Results on task 4 subtask A: Audio tagging (Instance-based evaluation)

	Baseline [11]	Our model
F-score	10.9	51.81
Precision	7.8	54.14
Recall	17.5	49.67

Table 6: Results on task 4 subtask B: Sound event detection (Segment-based overall metrics)

	Baseline [11]	Our model
ER	1.02	0.93
F-score	13.8	40.63

4. CONCLUSIONS

In this work, we proposed various deep learning model are formed form CNNs and RNNs for all tasks of DCASE 2017 challenge. Overall, our proposed models outperform the baseline system. For ASC task, we ob-tained an average accuracy of 82% compared to the baseline of 74.8%. For rare sound events detection problem, we achieved a mean error rate of 0.25 and F-score of 86.4%, which are a signifi-cant improvement in comparison with the baseline with error rate of 0.53 and F-score of 72.7%. For polyphonic sound events detec-tion, our approach reported a slight improvement in error rate of 0.59 and the baseline of 0.69. For task 4, we present a simple approach for the both two subtasks: audio tagging and sound event detection. Our model after learning from weakly labelled data can perform good detection.

5. REFERENCES

[1] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," *CoRR*, vol. abs/1609.04243, 2016. [Online]. Available: http://arxiv.org/abs/1609.04243

[2] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks," DCASE2016 Challenge, Tech. Rep., September 2016.

[3] Y. Han and K. Lee, "Convolutional neural network with multiple-width frequency-delta data augmentation for acoustic scene classification," DCASE2016 Challenge, Tech. Rep., September 2016.

[4] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," *CoRR*, vol. abs/1604.00861, 2016. [Online]. Available: http://arxiv.org/abs/1604.00861

[5] T. H. Vu and J.-C. Wang, "Acoustic scene and event recognition using recurrent neural networks," DCASE2016 Challenge, Tech. Rep., September 2016.

[6] E. Çakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *CoRR*, vol. abs/1702.06286, 2017. [Online]. Available: http://arxiv.org/abs/1702.06286

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[10] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[11] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017, submitted.