DCASE 2019 CHALLENGE TASK1 TECHNICAL REPORT

Technical Report

Houwei Zhu[1], Chunxia Ren[2], Jun Wang[2], Lei Yang[1], Shengchen Li[2,] Lizhong Wang[1]

[1] Samsung Research China-Beijing,Beijing, P. R. China
(houwei.zhu, lei81.yang, lz.wang)@samsung.com
[2] Beijing University of Posts and Telecommunications, Beijing, P. R. China
(chunxiaren, wangjun19930314, shengchen.li)@bupt.edu.cn

ABSTRACT

This report describe our methods for the DCASE 2019 task1a and task1c of Acoustic Scene Classification (ASC).Especially task1c for unknown scene, which not included in training dataset, We use unbalance unknown class training data and a threshold to classify the known and unknown scenes. In our method, we use Log Mel Spectrogram with different divisions, as the input of multiple neural network, the ensemble learning out-put shows good accuracy. For task 1a we use VGG and xception as network and 3 different divisions ensemble, the accuracy is 0.807 for Leadboard dataset. For task 1c we use Convolutional Recurrent Neural Network (CRNN) and self-attention mechanism with 2 different features division ensemble, and a threshold for unknown judgment, the Lead-board accuracy is 0.653.

Index Terms— audio scene classification, Log Mel-Spectrogram, CRNN, self-attention mechanism, ensemble

1. INTRODUCTION

The acoustic scene classification (ASC) is a complex problem but crucial in industrial, especially in home entertainment electronics, smart home, and complex environment automatic speech recognition. The IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) attracted many researches from academic and industry [1-4].

From previous DCASE challenge results, the deep learning, such as CNN [5], CRNN [6] and ensemble learning [7] shows great performance on this area. The audio data is mostly generated the time-frequency info, such as Mel Frequency Cepstral Coefficients (MFCC) [8], Constant Q transformation (CQT) [9], and other audio signal features [9]. From last year's top rank results, CNN network achieved good performance, RNN network shows better performance in time correlation signal.

We made different divisions of mel-spectrogram, use mean probabilities and ensemble. In this paper, we will describe the detail of our methods we pro-posed, experimental and conclusion.

2. PROPOSED METHOD

This section describe the audio feature extraction and neural network detail of our method. For task1a, we use 2 different network and 3 different features, total 6 networks ensemble. For task1c, 1 feature or 2 features are used, different threshold for unknown class judgement and self-attention based CRNN network was used construct 4 different final submissions.

2.1. Audio Features

We use Log Mel-spectrogram as audio feature. Most of DCASE 2018 top team use this as the feature [5][10], it shows better performance than MFCC, CQT, wavelet [11], HPSS [5] in our experiment.

Different time division log mel-spectrogram achieved different results. The DCASE 2019 dataset is 10 seconds 48000Hz audio, we use below Table.1 features different short-time Fourier transformation (STFT) window setting and different mel-spectrogram filter bins. The main frame length is 1920, 2048 and 4800 points. Then the STFT apply with Mel filter bank, the number of bandpass filters was 200,100 and 128. We use log mel-spectrogram because of the human ear is sensitive at low frequency and insensitive at high frequency. The 10 seconds audio has been divided into 9, 10, 5 and 1 segments as described in table.1.

In Task1a, the Acoustic Scene Classification dataset provided stereo audio for training and testing, the stereo audio contains spatial alternation information, and for example the vehicles run from left to right, we take advantage of the stereo audios. We also mixed left and right channel, it contains full information than individual channel info. For task1a we use 3 channels (Left, Right and Mixed) as input features. The 10 seconds audio chunk divided to 2 seconds segments and 1 second segments, then the 1 audio train data divide into 9 and 10 train data, it's kind of data augment.

In Task1c, the dataset provided mono channel audio, we use features 4 and 5 in Table 1.

In Task1a, features 1, 2, 3, 5(feature 5 with 3 channels) was used. In task1c, features 4, 5 was used.

Features Index	STFT-Window Size	Overlap	Channels	Time Division	Feature Shape (segs*channel*frame*mel-bin)
1	1920	1440	Left/Right/Mixed	2 sec len with 1 sec overlap, divided to 9 segs	9*3*200*200
2	1920	1440	Left/Right/Mixed	1 sec divided to 10 segment	10*3*100*100
3	4800	2400	Left/Right/Mixed	10 sec	1*3*200*200
4	2048	1301	Mixed	2 sec	5*1*128*128
5	2048	1052	Mixed	1 sec	10*1*48*128

Table 1: Multi division of the feature proposed by us

2.2. Network Architecture

2.2.1. CNN

The 1st network we used in task1a was VGG CNN network proposed by Kong [12], CNN8 in Kong's experiments in 2018 task1a, the results is 0.68 when use mono audio feature. We got 0.70 when use stereo audio, and 0.71 as 3 channels audio features in 2018 development set.

2.2.2. Xception

The 2nd network we used in task1a was Xception [13]. Xception is based on Depthwise separable convolution layer, we directly use this network and didn't made any change, the input shape is 3*200*200 and 3*100*100 for task1a.

2.2.3. Self-attention based CRNN

In this technical report, we use CRNN-based self-attention mechanism model [14] in task1c. In order to get more scene information, we use self-attention mechanism [15] to encode its input. Self-attention layer can capture global information well and solve the problem of long-distance dependence. RNN needs to be re-cursive step by step to get scene information and CNN relies on cascade to expand the field of receptivity. So we combine with self-attention mechanism after CRNN model. The model framework is as shown in the figure1: Self-attention based CRNN structure.

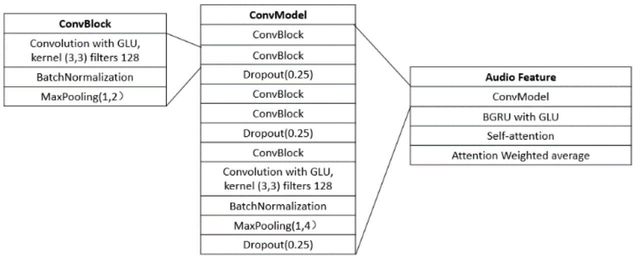

Figure 1: Self-attention based CRNN structure

2.3. Data Augmentation

In order to solve the problem of model over-fitting and scarcity of data sets, we used data enhancement methods—Mix-up [16] when training the model. Mixup extends the prior knowledge of training distribution by combining linear interpolation of feature vectors. Mixup builds a virtual sample in the following way

$$\hat{x} = \lambda x_i + (1 - \lambda) x_j \qquad (1)$$

$$\hat{y} = \lambda y_i + (1 - \lambda) y_j \qquad (2)$$

Where x_i, x_j are two samples randomly extracted from the raw training data set. The value of λ is between 0 and 1.yi; yj are one-hot label encodings.

The mix up random mix 2 samples (x_i, y_i) and (x_j, x_j) from the training data, According to our experiment, by using mix up, the accuracy increased about 1% in task1a and task1c. Furthermore, we try to mix 2 examples from same classes, other than random from all training data, we believe this will improve the generalization ability for this task. For this DCASE challenge, the training data is from 9 cities, and validation data is from 10 cities, the final evaluation data is from 12 cities, we try to mix different cities same scene audio clips, we believe it will improve the performance for the data predict from not include in training data cities.

3. EXPERIMENT AND RESULTS

3.1. Dataset

TAU Urban Acoustic Scenes 2019 dataset is used in this report. For task1a TAU Urban Acoustic Scenes 2019 dataset include 10 classes' acoustic scenes from 10 cities total 14400 segments stereo audio files. The default train/test set is 9185 in train (not include audio from Milan), 4185 in test, additional 1030 segments from Milan. We use the default setting for training. The evaluation dataset is from 12 cities (2 cities not encountered in development set).

For task1c use TAU Urban Acoustic Scenes 2019 Openset dataset, and some additional data as "unknown" acoustic scenes. The development dataset include 14400 of ten scene classes as above task1a mix mono channel audio, and 1450 unknown class segments, actually the 1450 unknown segments include 4 classes. The evaluation dataset include 10 known classes and part in other unknown environments. We use the officially divided training set, where unknown class has three scenes, and there is another unknown scene in the official test set.

3.2. Task1a Results

In the table below, we compare our network accuracy and the official baseline accuracy for Task1a.(Some data results are not given in the technical report due to time.)

NN type	**Feature choose**	**Results (%)**	**Leadboard Result**
Base line		64.33	
Xception	Feature 1	-	
	Feature 2	-	
	Feature 3	-	
VGG CNN	Feature 1	-	
	Feature 2	-	
	Feature 3	-	
Attention based CRNN	Feature 4 * 3channel	73.10	
Ensemble	Mean ensemble	-	
Lead-board			80.67

Table 2: Overall accuracy in Task 1a

3.3. Task1c Results

In the table below, we compare our network correctness rate and the official correct rate for Task1c.

Considering that the officially released data set has undefined unknown class, so, when the system outputs the correct probability of each audio segment, we add a threshold judgment mechanism, that is, when the audio scene output probability is lower than 0.4, it is determined as unknown class.

NN type	Feature choose	Results (%)	Leadboard Result
Base line		46.67	46.67
Attention based CRNN	Feature 4	58.31	-
Attention based CRNN	Feature 5	50.69	-
Ensemble	Mean ensemble	57.27	65.33

Table 3: Overall accuracy in Task 1c

3.4. Experiment setting

The parameters including learning rate, decay, momentum and initial training weights were default provided by baseline Keras. We use Adam as stochastic gradient descent for optimizer of the network.

Training and validation data was default setting by DCASE organizer. We pre-train for several hours for each network. For Leadboard and evaluation, we use all dev data as training data, and default validation data as validation data, load the pre-trained weight for each network, it need to train for less than 10 epochs when we use batch size as 128, to prevent the overfitting problem.

For task1c we use both 0.4 and 0.41 as a threshold, in different submissions, when the predict probability of our network for the audio segment is less than the threshold, we judge it as unknown class for the predict result.

4. CONCLUSION

The report conclude our methods for DCASE task1a and task1c. For task1a we considered generalization ability of our system is a significant factor, because the default training data is from 9 cities, but the evaluation data is from 12 cities, we tried data augment by using mix-up, and modified mix-up, we believe same class mix up with different cities should improve the generalization ability.

For task1c unknown scene is a challenge, we only use 11 classes classification similar with task1a, but it's regular problem in image and face recognition fields, in future we plan to pay more attention on the unknown judgement. It's practical on industrial.

5. REFERENCES

[1] http://dcase.community/workshop2019/.

[2] Mun S, Park S, Han D K, et al. Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane[J]. Proc. DCASE, 2017: 93-97.

[3] Lim H, Park J, Han Y. Rare sound event detection using 1D convolutional recurrent neural networks[C]//Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop. 2017: 80-84.

[4] Adavanne S, Virtanen T. A report on sound event detection with different binaural features[J]. arXiv preprint arXiv:1710.02997, 2017.

[5] Sakashita Y, Aono M. Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions[J]. IEEE AASP Challenge on DCASE 2018 technical reports, 2018.

[6] Kukanov I, Hautamäki V, Lee K A. Recurrent neural network and maximal figure of merit for acoustic event detection[J]. IEEE AASP Challenge on DCASE 2017 technical reports, 2017.

[7] Han Y, Park J, Lee K. Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification[J]. the Detection and Classification of Acoustic Scenes and Events (DCASE), 2017: 1-5.

[8] Dorfer M, Lehner B, Eghbal-zadeh H, et al. Acoustic scene classification with fully convolutional neural networks and I-vectors[J]. IEEE AASP Challenge on Detection and Classification of Acoustic Scen and Events (DCASE), 2018.

[9] Zeinali H, Burget L, Cernocky J. Convolutional neural networks and x-vector embedding for dcase2018 acoustic scene classification challenge[J]. arXiv preprint arXiv:1810.04273, 2018.

[10] Zhang L , Han J . Acoustic scene classification using multi-layer temporal pooling based on convolutional neural network[J]. 2019.

[11] Qian K, Ren Z, Pandit V, et al. Wavelets revisited for the classification of acoustic scenes[C]//Proc. DCASE Workshop, Munich, Germany. 2017: 108-112.

[12] Kong, Qiuqiang, Turab Iqbal, Yong Xu, Wenwu Wang, and Mark D. Plumbley. "DCASE 2018 Challenge baseline with convolutional neural networks." arXiv preprint arXiv:1808.00773 (2018)

[13] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[14] Wang J, Li S. Self-Attention Mechanism Based System for Dcase2018 Challenge Task1 and Task4[J]. IEEE AASP Challenge on DCASE 2018 technical reports, 2018.

[15] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.

[16] Zhang H , Cisse M , Dauphin Y N , et al. mixup: Beyond Empirical Risk Minimization[J]. 2017.