AN AUDIO SCENE CLASSIFICATION FRAMEWORK WITH EMBEDDED FILTERS AND A DCT-BASED TEMPORAL MODULE

Hangting Chen[1,2] *Pengyuan Zhang*[*1,2] *Yonghong Yan*[1,2,3]

[1]Key Laboratory of Speech Acoustics & Content Understanding, Institute of Acoustics, CAS, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Xinjiang Laboratory of Minority Speech and Language Information Processing,
Xinjiang Technical Institute of Physics and Chemistry, CAS, China

ABSTRACT

Deep convolutional neural network (DCNN) has recently improved the performance of acoustic scene classification. However, the input features of the network are usually based on predefined hand-tailored filters, which may not apply to the specific tasks. To overcome this, we propose a hybrid framework that jointly trains the front-end filters and the back-end DCNN. Also, a novel temporal module based on the discrete cosine transform (DCT) is inserted after the high-level feature map of the network, thus enabling us to utilize time information without a reduction of training samples. Our single system, composed of the fine-tuned wavelet front-end and the DCNN back-end, with the integrated DCT-based temporal module, has achieved an accuracy of 79.20% in the evaluation set in DCASE17, gaining around 3% and 8% accuracy improvement compared with scalogram-DCNN and FBank-DCNN systems, respectively.

Index Terms— Acoustic scene classification, embedded filters, joint-training, DCT-based temporal module

1. INTRODUCTION

Environmental sound contains a large amount of surrounding information. Acoustic scene classification (ASC) aims to classify the sound into one of predefined classes, e.g., park, office [1], with applications for context-aware devices [2].

One of the biggest challenges is to find proper acoustic features due to the various time scales of environmental information. Most features are based on short-time Fourier transform (STFT), for example the Mel-scale filter bank (FBank) [3]. The scalogram, which represents the spectrum of constant-Q wavelet transform, has gradually gained more attention for its ability to sense signals at different time scales [4][5][6][7]. However, all these filters are hand-tailored based on perceptual research. Recently, some studies have made an attempt to build systems directly from the raw waveform [8][9][10] and part of them have achieved state-of-the-art performance [11]. Yet the limited data size and various network topology make the task quite difficult. In this study, a novel joint-training neural network is proposed, which integrates heuristic knowledge and the potential of deep learning. The Mel and wavelet filters are embedded into a front-end module and jointly trained with a back-end deep convolutional neural network (DCNN) module.

Another concern is the mismatch of the segment-level label and frame-wise training. Given a segment-level label, the general method is to assign each frame the target label [12]. Temporal pooling layers, for example, the mean pooling [6] and attention [13], summarize the whole feature map into one vector, which reduces the fluctuation with little loss of crucial information. However, for the ASC task, the environmental sounds do not necessarily exhibit strong temporal dynamics [14]. The background information stored in each frame is usually sufficient for classification. Moreover, frame-wise prediction is more efficient and flexible for real life application. In this paper, we present a novel temporal module based on discrete cosine transform (DCT), which filters the high-level feature map while keeping frame-wise information.

As much research has already been conducted in the ASC task, the unique contribution of this work is two-fold. First, the joint-training framework optimizes both the front-end feature extraction module and back-end DCNN. To the best of our knowledge, this is the first attempt to embed predefined complex filters into the network with various joint-training settings. Second, the DCT-based temporal module serves as a simple but effective approach utilizing time information without the reduction of training samples. Noting that the mentioned two approaches are not limited to the ASC task, they can be deployed for the features extracted from spectrogram and tasks related to sequence labeling.

On the evaluation set, integrated with the DCT-based temporal module, the "Mel" filters trained from scratch have achieved a 3.09% accuracy improvement compared with the original FBank-DCNN system. Likewise, the fine-tuned

This work is partially supported by the National Natural Science Foundation of China (Nos.U1536117,11590770-4),the Pre-research Project for Equipment of General Information System (No.JZX2017-0994/Y306).

* Pengyuan Zhang is the corresponding author.

wavelet filters has achieved a 3.15% accuracy improvement compared with the original scalogram-DCNN system.

The remainder of the paper is organized as follows. Section 2 and 3 introduce the proposed joint-training framework and the DCT-based temporal module. Section 4 describes the methodology of experiments and Section 5 presents the results. Section 6 discusses and concludes this work.

2. JOINT-TRAINING OF DCNN WITH EMBEDDED FILTERS

In this section, we first describe the scalogram-DCNN system which has demonstrated impressive performance improvement in the ASC task[15]. Then a joint-training framework is introduced to adjust filters from heuristic knowledge to task specification automatically.

2.1. Scalogram-DCNN system

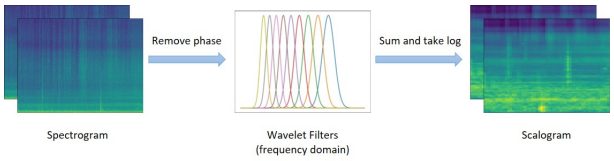

Fig. 1. The front-end feature extraction module. The raw waveform is transformed into a spectrogram and then filtered by wavelet filters. The scalogram is represented as the log of the summation of energy in each filter. If the filters are Mel, the extracted feature is FBank.

The extraction of the scalogram requires convolutional operations. In this paper, for simplicity and to be consistent with the FBank extraction procedure, the wavelets, which each act as a pass-band filter, operate on the spectrogram to generate a scalogram (Fig. 1).

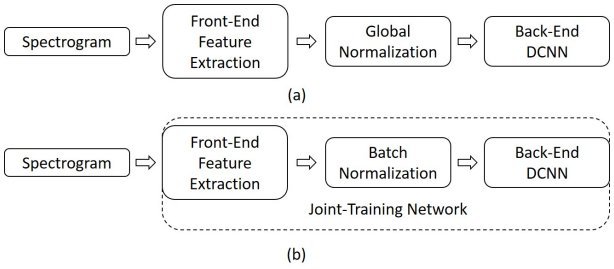

Fig. 2. (a) Classic procedure of classification from feature extraction to DCNN. (b) The joint-training framework which combines front-end and back-end module into one network.

In the back-end DCNN (Table 1), convolutional layers with small kernels are deployed to learn high-level features [16]. Note that the convolutional and pooling layers only operate on frequency/scale axis.

Table 1. The back-end DCNN. The input feature map is of size frames(L) $\times$ channels(c) $\times$ filters(n). The notation "$c\times3$ Conv(pad-0,stride-1)-$2c$-BN-ReLu" denotes a convolutional kernel with c input channels, $2c$ output channels and a size of 3, followed by batch normalization and ReLu activation.

Layer Name	Settings
Input	Spectrum $L \times c \times n$
Conv1	$c \times 3$ Conv(pad-0,stride-1)$\times 2c$-BN-ReLu 2 Pooling(pad-1,stride-2)
Conv2	$2c \times 3$ Conv(pad-0,stride-1)$\times 4c$-BN-ReLu 2 Pooling(pad-0,stride-2)-Dropout
Conv3	$4c \times 3$ Conv(pad-0,stride-1)$\times 8c$-BN-ReLu 2 Pooling(pad-0,stride-2)
Conv4	$8c \times 3$ Conv(pad-0,stride-1)$\times 16c$-BN-ReLu 2 Pooling(pad-0,stride-2)-Dropout
	Concatenate and flatten input as well as Conv's output
FC1	Linear (128 units)-BN-ReLu-Dropout
FC2	Linear (128 units)-BN-ReLu-Dropout
FC3	Linear (128 units)-BN-ReLu
Output	15-way SoftMax

2.2. The joint-training framework

The filter used to extract acoustic features is often predefined, representing human perceptual knowledge for acoustic modeling. Yet the settings of the filter may not be suitable for the specific task.

The joint-training framework combines the front-end feature extraction module and the back-end DCNN module(Fig. 2(b)). As the Mel and wavelet filters are defined in the frequency domain, they can be conveniently embedded into the framework's front-end part. Batch normalization [17] is then applied to the extracted features, followed by the back-end DCNN module. Nevertheless, the training is not straightforward at first sight, for the parameter size of filters is equal to or even larger than the back-end DCNN system in this case. Another concern is that the filters need to be sparse and orthogonal to some extent, suggesting that different filters should be located at different parts on the frequency axis. In our design, each filter's covering range is recorded and parameters only cover a small part of the frequency domain. For Mel filters, the start and end indexes are set to the vertex of the triangle base. For wavelets, the indexes are determined by maintaining its constant-Q property, that is, the center frequency and bandwidth are distributed in a logarithmic scale. Meanwhile, this design dramatically saves memory and reduces the number of parameters to be trained.

The joint-training framework integrates heuristic knowledge of the filters' location and performs a direct flow from the spectrogram to frame-wise posterior probability. The batch normalization in the joint-training framework fills the gap between the joint-trained features and the DCNN input. In contrast, for classic training procedures, the acoustic features are pre-calculated by the fixed filters and the global normalization coefficients are determined within the dataset.

3. DCT-BASED TEMPORAL MODULE

The proposed DCT-based temporal module aims to utilize information along time in frame-wise training and prediction methods (Fig. 3(c)). The input feature map is split into non-overlapping chunks $X \in R^{T \times N}$, where T is the number of frames in the chunk, N is the dimension of the feature. An additional variance map Y is then calculated along time. After DCT, weighted by learnable matrices W_X and W_Y, the inverse discrete cosine transform(IDCT), the X and Y changes to $\tilde{X}$ and $\tilde{Y}$. An additional linear transform follows.

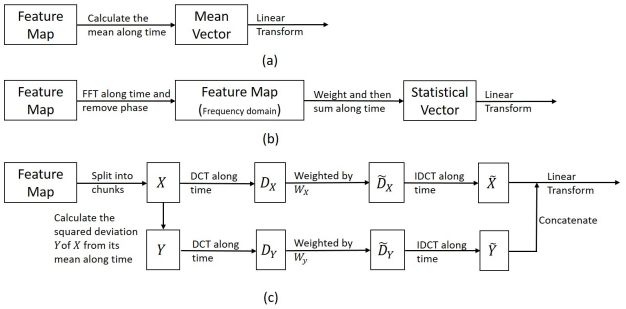

Fig. 3. The mentioned temporal modules. (a) Mean pooling. (b) Temporal transformer module. (c) DCT-based temporal module.

Inspired by the temporal transformer module [18], the proposed module has made three main changes. First, an additional variance feature map is added to depict second-order characteristic. Second, DCT is applied on the chunk instead of discrete Fourier transform (DFT) on the whole segment. DCT deals with the temporal shifting as well as DFT in [18], but completely in the real value field. The chunk design reduces the size of weight matrices. In addition, the weight matrix can be regarded as a filter, providing an attention weight on the DCT spectrum. Third, in contrast with the temporal transformer summarizing the whole input feature map into one vector, the proposed temporal module recovers the signal to support frame-wise training and prediction.

4. EXPERIMENTAL SETUP

4.1. Dataset

Our experiments were conducted on the dataset of DCASE17 Task 1 [19], which includes development (Dev.) and evaluation (Eva.) part, with a total of 15 acoustic scenes. The officially provided 4-fold cross validation procedure was used to tune hyper-parameters. The performance of proposed systems was evaluated by the mean accuracy of 4-fold cross validation on the development dataset (CV. on Dev.), and the accuracy on the evaluation dataset (Acc. on Eva.). For early stopping, around 8% of the training samples were selected for validation.

4.2. Joint-training procedure

To explore the performance of joint training, two types of filters were used, the Mel and wavelet filters. For Mel filters, STFT was applied on the raw signal every $20ms$ over $40ms$ windows. The total number of triangle filters was 40, covering $10Hz$ to $22.05kHz$. The FBank of a 10-second stereo audio was of the dimension $500 \times 2 \times 40$ and $500 \times 6 \times 40$ after adding delta and delta-delta coefficients. For wavelets, STFT was applied on the raw signal every $185ms$ over $555ms$ windows. The total number of wavelets was 92, distributed uniformly at low frequency and logarithm at high frequency as described in [20]. The scalogram of a 10-second stereo audio was of the dimension $54 \times 2 \times 92$. We followed the wavelet settings in our previous work [15]. Furthermore, the covering frequency range of each filter was recorded down, and only the weight in the range was registered as the parameter embedded in the font-end.

Three methods were experimented individually, original, tuned and training from scratch. In this paper, the notation "{Mel,wavelet}-{original,tuned,scratch}" denotes the type of the filters and training methods. The classic approach used original filters extracting acoustic features to training DCNN models, as in Fig. 2(a). The joint training was carried on Fig. 2(b). For {Mel,wavelet}-scratch, the filters were initialized by sampling from the normal distribution $\mathcal{N}(1.0, 0.1^2)$ and the whole network was trained together. For {Mel,wavelet}-tuned, the filters were initialized by original filters and fixed. Only the back-end network was tuned at first, followed by fine-tuning the whole network using a smaller learning rate.

Table 2. Results of experiments of different training frameworks. The best performance for each filter type is in bold.

Filter	Method	CV. on Dev.(%)	Acc. on Eva.(%)
Mel	Original	81.03	71.11
Mel	Tuned	**81.67**	71.85
Mel	Scratch	81.39	**73.02**
Wavelet	Original	86.45	76.05
Wavelet	Tuned	**87.15**	**77.16**
Wavelet	Scratch	83.70	72.04

4.3. Temporal module settings

In this study, the mean pooling, temporal transformer module and DCT-based temporal module were tested. All of them were inserted between the linear transform and batch normalization in FC3 of DCNN (Table 1). The output of linear transform in temporal modules was of dimension 128. The chunk size was set to 25 in FBank and 18 in scalogram.

4.4. Model training

The training was frame-wise and the prediction of the posterior probability was obtained by averaging all frames' log-posterior. We used Adam [21] to update parameters, set the

Table 3. Experimental results of temporal modules on various filters, which are listed as CV. on Dev(%)/Acc. on Eval.(%). The best performance for each filter is in bold font. Note that DCT-based Temporal Module* did not use variance feature map.

Filter	Temporal Module			
	Mean Pooling	Temporal Transformer Module	DCT-based Temporal Module*	DCT-based Temporal Module
Mel-Original	**82.19**/71.85	81.37/**73.02**	82.08/72.10	81.93/72.90
Mel-Tuned	81.69/72.22	81.74/71.91	81.89/72.65	**82.19**/**73.33**
Mel-Scratch	82.53/72.84	81.33/71.85	82.61/73.52	**82.69**/**74.20**
Wavelet-Original	86.88/73.46	86.00/72.72	87.28/**74.94**	**87.33**/**74.94**
Wavelet-Tuned	87.26/77.35	86.34/74.81	87.39/78.83	**87.43**/**79.20**
Wavelet-Scratch	84.02/72.65	83.50/71.30	83.95/74.20	**84.06**/**74.44**

dropout rate [22] to 0.5, the initial learning rate to 10^{-3}, the initial fine-tuning learning rate to 5×10^{-4}. The learning rate shrink and training process termination followed the loss in the validation set. All training settings were unchanged for each part of experiments.

5. RESULTS

As shown in Table 2, two types of filters together with three methods were deployed individually to explore joint-training framework. The Mel-original and wavelet-original with the DCNN back-end served as baseline systems in this paper. For wavelets, the wavelet-tuned achieved the best performance, with an accuracy improvement of 0.70% on CV. and 1.11% on Eva., in contrast with original filters. The performance gap between the scratch and tuned was large. For Mel filters, the Mel-tuned and Mel-scratch both outperformed the original filters remarkably.

Moreover, under the same framework, the wavelets largely outperformed Mel filters with the exception of the from-scratch method, indicating that for modeling acoustic scenes, background information plays a crucial role and long-term filters have superiority over short-term ones.

The results of several back-end temporal modules are listed in Table 3. The temporal transformer module slightly benefited the model in our experiment. The mean pooling and DCT-based module (without the variance feature map) improved the performance and the latter achieved higher accuracy in most cases. Adding the variance feature map further improved the system's performance.

6. DISCUSSION

In the joint-training experiments, the fine-tuned filters always outperformed the original filters, demonstrating that filters optimized by the specific task can perform better in the task. The remarkable performance of the Mel-scratch could be attributed to the large size of training frames in FBank compared with long-term scalogram. In addition, the wavelets had more parameters, around 38 times that of Mel filters in our settings. Therefore, with more training samples and less learning parameters, the Mel-scratch filters achieved a competitive performance.

The Mel filters were chosen for visualization. The Mel-{original, tuned, scratch} filters and corresponding FBank in an audio are plotted in Fig. 4. The shape of Mel-tuned filters appears similar to that of the original, but not smooth. The Mel-scratch filters seem unreasonable from its visualization. But, to our surprise, its FBank looks almost equal to the other two, indicating that the neural network has the ability to automatically learn the filters with sufficient training samples.

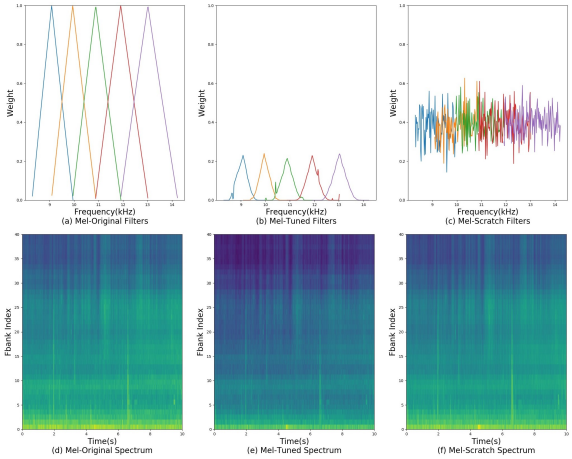

Fig. 4. Visualization of Mel filters and the corresponding FBank feature.

The DCT-based temporal module (with and without variance feature) achieved better performance. The other two temporal modules were trained in a segment-wise way. For mean pooling, the parameter size of the linear transform was around 4 times the number of training samples, not to speak of the temporal transformer module. However, the proposed DCT-based temporal module filtered the time information without reducing the number of training samples.

In conclusion, our final single systems, composed of Mel-scratch and wavelet-tuned front-end and DCNN back-end integrating the DCT-based temporal module, have achieved accuracy improvements of 3.09% and 3.15% in the evaluation dataset compared with {Mel,wavelet}-original systems, far exceeding the official baseline [19]. The improvement was not significant, but consistent. System and feature fusion strategies may further give rise to the performance.

7. REFERENCES

[1] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "Tut database for acoustic scene classification and sound event detection," in *Signal Processing Conference*, 2016, pp. 1128–1132.

[2] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Transactions on Audio Speech & Language Processing*, vol. 14, no. 1, pp. 321–329, 2006.

[3] J. Volkmann, S. S. Stevens, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *J.acoust.soc.am*, vol. 8, no. 3, pp. 185–190, 1937.

[4] Qian Kun, Ren Zhao, Pandit Vedhas, Yang Zijiang, Zhang Zixing, and Schuller Björn, "Wavelets revisited for the classification of acoustic scenes," Tech. Rep., DCASE2017 Challenge, September 2017.

[5] Zheng Weiping, Yi Jiantao, Xing Xiaotao, Liu Xiangtao, and Peng Shaohu, "Acoustic scene classification using deep convolutional neural network and multiple spectrograms fusion," Tech. Rep., DCASE2017 Challenge, September 2017.

[6] Hossein Zeinali, Lukas Burget, and Honza Cernocky, "Convolutional neural networks and x-vector embedding for dcase2018 acoustic scene classification challenge," Tech. Rep., DCASE2018 Challenge, September 2018.

[7] Ren Zhao, Kun Qian, Zixing Zhang, Vedhas Pandit, Alice Baird, and BjöRn Schuller, "Deep scalogram representations for acoustic scene classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, 2018.

[8] Jee-weon Jung, Hee-soo Heo, Hye-jin Shim, and Hajin Yu, "DNN based multi-level features ensemble for acoustic scene classification," Tech. Rep., DCASE2018 Challenge, September 2018.

[9] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das, "Very deep convolutional neural networks for raw waveforms," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 421–425.

[10] Pegah Ghahremani, Hossein Hadian, Hang Lv, Daniel Povey, and Sanjeev Khudanpur, "Acoustic modeling from frequency domain representations of speech," in *Proc. Interspeech 2018*, 2018, pp. 1596–1600.

[11] Mousmita Sarma, Pegah Ghahremani, Daniel Povey, Nagendra Kumar Goel, Kandarpa Kumar Sarma, and Najim Dehak, "Emotion identification from raw speech signals using dnns," in *Proc. Interspeech 2018*, 2018, pp. 3097–3101.

[12] Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[13] Fei Tao and Gang Liu, "Advanced lstm: A study about better time dependency modeling in emotion recognition," 2017.

[14] Juncheng Li, Wei Dai, Florian Metze, Shuhui Qu, and Samarjit Das, "A comparison of deep learning methods for environmental sound detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[15] Hangting Chen, Pengyuan Zhang, Haichuan Bai, Qingsheng Yuan, Xiuguo Bao, and Yonghong Yan, "Deep convolutional neural network with scalogram for audio scene modeling," in *Proc. Interspeech 2018*, 2018, pp. 3304–3308.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105.

[17] Sergey Ioffe and Christian Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," pp. 448–456, 2015.

[18] Teng Zhang, Kailai Zhang, and Ji Wu, "Temporal transformer networks for acoustic scene classification," in *Proc. Interspeech 2018*, 2018, pp. 1349–1353.

[19] Toni Heittola and Annamaria Mesaros, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," Tech. Rep., DCASE2017 Challenge, September 2017.

[20] Joakim Andén and Stéphane Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, 2014.

[21] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.