CONVOLUTIONAL NEURAL NETWORK WITH MULTIPLE-WIDTH FREQUENCY-DELTA DATA AUGMENTATION FOR ACOUSTIC SCENE CLASSIFICATION

Yoonchang Han

Music and Audio Research Group,
Seoul National University,
Seoul, Korea
yoonchanghan@snu.ac.kr

Kyogu Lee[*]

Music and Audio Research Group,
Seoul National University,
Seoul, Korea
kglee@snu.ac.kr

ABSTRACT

In this paper, we apply convolutional neural network on acoustic scene classification task of DCASE 2016. We propose multi-width frequency-delta data augmentation which uses static mel-spectrogram as well as frequency-delta features as individual examples with same labels for the network input, and the experimental result shows that this method significantly improves the performance compare to the case of using static mel-spectrogram input only. In addition, we propose folded mean aggregation, which first multiplies output probabilities of static and delta augmentation data from the same window first prior to audio clip-wise aggregation, and we found that this method reduces the error rate further. The system exhibited a classification accuracy of 0.831 on the development set and 0.846 on the evaluation set.

***Index Terms*—** DCASE 2016, acoustic scene classification, convolutional neural network, deep learning, multi-width frequency-delta data augmentation

1. INTRODUCTION

In recent years, a number of internet of things (IoT) devices and intelligent personal assistants (IPAs) applications are released. Many of these devices and applications aim to provide service at an appropriate time, thus understanding 'context' become more and more important. In machine listening field, recognizing the sound event and sound scene are important goals to understand the context and environment that surrounding users. However, the research community has lacked benchmark dataset to compare algorithms so far [1], thus IEEE Audio and Acoustic Signal Processing (AASP) Technical Committee organized Detection and Classification of Acoustic Scenes and Events (DCASE) challenge in 2013 which covers acoustic scene classification (ASC) and sound event detection tasks. The objective of the ASC task of DCASE 2013 was classifying 10 acoustic scenes and 10 audio segments were provided for each class. The length of each segment was 30 s, total 11 algorithms are submitted for the challenge [2], and hand-crafted features such as spectral/temporal features along with classifier are the most popular choice.

In 2016, IEEE AASP organized another challenge, DCASE 2016, with new audio recordings and extended dataset in terms of variety and volume. In the past 3 years, various deep neural network approach has proposed for audio processing and have outperformed conventional hand-crafted features. Deep learning approach requires a sufficiently large amount of input data to learn a good feature representation, and the new DCASE 2016 dataset contains 15 acoustic scenes and 78 audio segments per scene which is much larger than the previous challenge and gives us an opportunity to apply deep learning approach.

In this paper, we demonstrate how we applied convolutional neural network (ConvNet) which effectively learns distinctive local characteristics from the input data [3], time-frequency representation in this case. We propose input data augmentation and output probability aggregation method which are explained in the next section.

2. PROPOSED SYSTEM

2.1. DCASE 2016 Dataset

The ASC dataset of DCASE 2016 includes 15 scenes which are bus, cafe/restaurant, car, city center, forest path, grocery store, home, lakeside beach, library, metro station, office, residential area, train, tram, and urban park. 78 audio segments were provided per scene, and each audio segment is 30 s audio clip and it is recorded using Soundman OKM II Klassik/studio A3, an electret binaural microphone[1] and a Roland Edirol R-09 wave recorder[2] under 44,100 sampling rate with 24-bit resolution. The organizer provided 4-fold cross validation setting and the baseline system which uses mel-frequency cepstral coefficients (MFCCs) and its delta/double delta, then classified with gaussian mixture models (GMMs). More details about the dataset and baseline system can be found in [4]. We used provided development set for experiments and result using evaluation set is explained in the Chapter 4.

2.2. Audio Preprocessing

First, we converted stereo audio to mono by taking mean values from the left and right channel. We used full 44,100 Hz sampling frequency without downsampling because it seemed meaningful spectral characteristics observed in a high-frequency range from the visual inspection on the spectrogram. Then we divided 30 s into 1 s audio chunks for training and testing without window overlapping following [5] which uses similar ConvNet architecture for the musical instrument identification task. For time-frequency representation of the audio, we used mel-spectrogram with 128 bin which is a sufficient size to keep spectral characteristics while greatly reduces the feature dimension. The window size used for short-time Fourier

[*]K. Lee is also with the Advanced Institutes of Convergence Technology, Suwon, Republic of Korea

[1]http://www.soundman.de/en/products/

[2]http://www.rolandus.com/products/r-09/

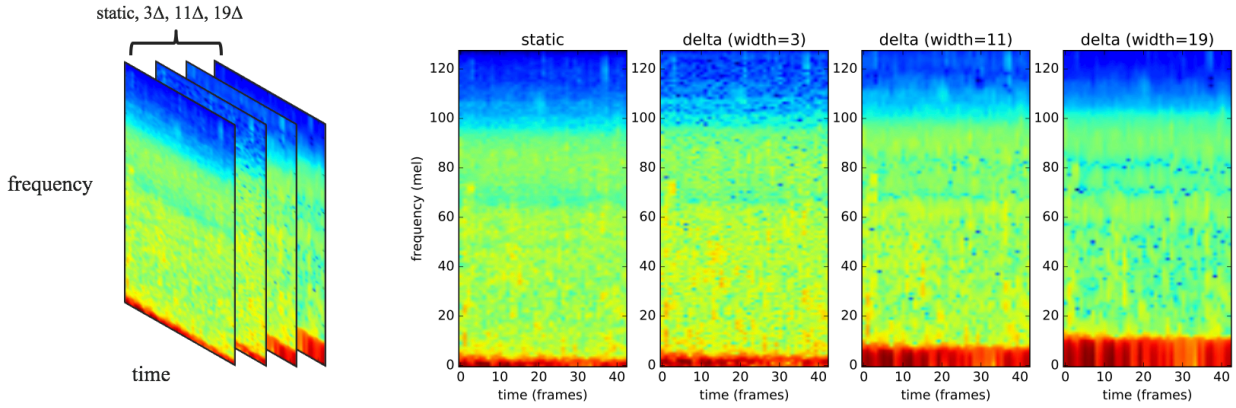

Figure 1: Two different ConvNet input organization method. (a) is a typical method that using several feature maps for ConvNet input. (b) is proposed MWFD data augmentation which uses frequency-delta features and feeds all to ConvNet as individual examples along with the static input with the same labels.

transform was 2048 samples with the hop size of 1024 which are approximately 46 ms and 23 ms, respectively. Finally. we standardized the values prior to feed them into the network by subtracting the mean and divide them with the standard deviation. Note that we obtained mean and standard deviation statistics from the training set only, and testing set was standardized using the statistics obtained from the training data.

2.3. Network Architecture

We formed a ConvNet architecture using 8 convolution layers with max-pooling after every two convolution layers. We used repeated small 3×3 receptive field, inspired by VGGNet [6] and overall architecture is presented in 1. We increased the number of filters every two convolution layer from 32 to 256 and added zero-padding before convolution layers to make full use of values near surrounding edges. For activation function, instead of widely used rectified linear unit (ReLU), we used leaky ReLU which is proposed by Mass et al. [7] because it is reported that it effectively reduces the error rate [5] by preventing 'dead' activation of initially inactive units. Unlike normal ReLU suppress all negative part to zero, leaky ReLU gives small gradient to the negative part and it is defined as:

$$y_i = \begin{cases} z_i & z_i \geq 0 \\ \alpha_i z_i & z_i < 0 \end{cases} \quad (1)$$

where α between 0 and 1 decides the slope on the negative part and set as 0.33 in our network which can be considered as 'very leaky' setting. Leaky ReLU was applied on all convolution layers as well as fully-connected layer prior to the classification layer which uses softmax activation. After the final convolution layer, we performed global average pooling prior to feeding features into the fully-connected layer. The network weights are initialized with Glorot uniform and used stochastic gradient descent (SGD) with Nesterov momentum for the optimizer. Learning rate was set as 0.02 with early stopping patience of 15 epochs and validation set for early stopping was randomly chosen 15% the training data.

Data shape	Description
$1 \times 43 \times 128$	mel-spectrogram
$32 \times 45 \times 130$	3×3 convolution, 32 filters
$32 \times 47 \times 132$	3×3 convolution, 32 filters
$32 \times 15 \times 44$	3×3 max-pooling
$32 \times 15 \times 44$	dropout (0.25)
$64 \times 17 \times 46$	3×3 convolution, 64 filters
$64 \times 19 \times 48$	3×3 convolution, 64 filters
$64 \times 6 \times 16$	3×3 max-pooling
$64 \times 6 \times 16$	dropout (0.25)
$128 \times 8 \times 18$	3×3 convolution, 128 filters
$128 \times 10 \times 20$	3×3 convolution, 128 filters
$128 \times 3 \times 6$	3×3 max-pooling
$128 \times 3 \times 6$	dropout (0.25)
$256 \times 5 \times 8$	3×3 convolution, 256 filters
$256 \times 7 \times 10$	3×3 convolution, 256 filters
$256 \times 1 \times 1$	global average-pooling
1024	flattened and fully connected
1024	dropout (0.50)
15	softmax

Table 1: Proposed ConvNet architecture. The data shape indicates (number of filters $\times$ time $\times$ frequency). The activation functions and zero-paddings are not shown for brevity.

2.4. Multi-width Frequency-delta Data Augmentation

To increase the classification performance of ConvNet further, we propose multi-width frequency-delta (MWFD) data augmentation which is illustrated in Fig.1. This method uses delta features in a frequency axis with a several different delta widths to emphasize spectral characteristics in a various resolution. We obtained MWFD

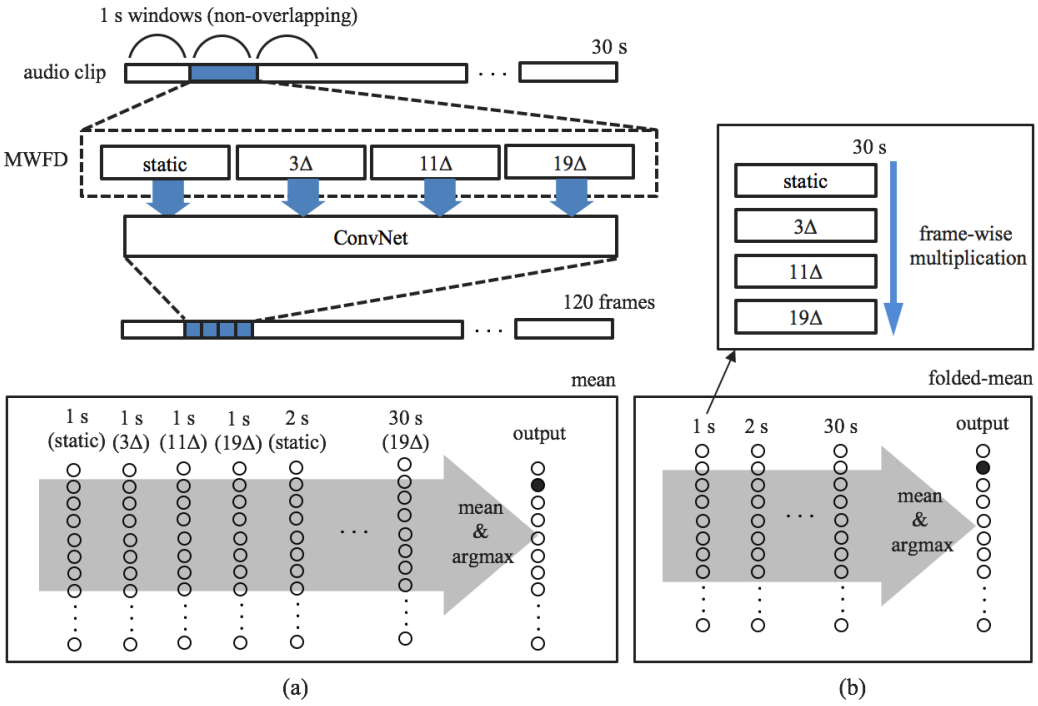

Figure 2: Overall system architecture with MWFD data augmentation and two different aggregation methods. (a) illustrates the case of taking an average of all 120 softmax outputs per audio clip (*S1*). (b) demonstrates aggregation of static and MWFD data from same input audio window by frame-wise multiplication first, then take an average over them (*S2*).

features in a same manner with calculating temporal delta feature of MFCCs. Frequency delta is defined as:

$$d_f = \frac{\sum_{k=1}^{K} k(x_{f+k} - x_{f-k})}{2\sum_{k=1}^{2} k^2} \quad (2)$$

where d_f is obtained delta feature at frequency bin f, and the only difference from delta MFCCs is that it is calculated along the frequency axis. K decides the range of frequency bins covered for delta feature calculation and the term 'delta width' across the paper indicates $2K + 1$. In the experiment, we used delta widths of 3, 11, and 19 which covers reasonably wide frequency range while extracting meaningfully different features. For delta calculation on the edges, we simply padded the data with repeated edge values prior to delta calculation to keep the data size same.

Usually, it is common to put several different versions of the input data as feature maps for ConvNet. For instance, color image input for ConvNet usually uses 3 feature maps, each contains information about the magnitude of the red, green, and blue color of the image which share the same local region [8]. However, MWFD features contain the information about emphasized spectral characteristics in a various resolution which can be considered as edge-emphasized versions of the input rather than containing independent information like a color.

Deep ConvNet architecture basically aims to learn features that are suitable to describe the input data. The first layer usually learns very simple shapes such as horizontal, diagonal, or vertical edges and following layers use these learned features as a component to learn higher level features. Hence, we think that putting all static feature and MWFD features as individual examples would be more helpful for ConvNet to extract appropriate edge-like features rather than putting them as feature maps as for computer vision tasks. In the experiment, we compare 3 different input arrangement settings which are static input only, put MWFD as feature maps, and put MWFD as individual examples (MWFD spread).

2.5. Folded Mean Aggregation

Because we use static input as well as 3 MWFD features as individual examples for ConvNet, the number of input and output probability become 4 times larger compare to the case using only static input. This means that 30 s audio clip produces 120 output probabilities with MWFD augmentation while static only case 30 generates 30 output probabilities.

To make the most of MWFD feature augmentation, we propose folded mean aggregation which multiplies output probabilities of static and MWFD features from the same window first. This process summarizes 120 output probabilities into 30 output probabilities. In the experiment, we compare two aggregation methods which are taking average over all 120 output probabilities ($S1$), and the proposed folded mean aggregation ($S2$) as illustrated in Fig.2

3. RESULTS

We used 4-fold cross-validation setting provided by the organizer. We repeated all experiments 3 times per fold to obtain the mean accuracy but with 3 different random seeds which are kept same across the algorithms to make experiment fair as possible. As shown in Fig.2, plain version of proposed ConvNet with static mel-

Algorithms	Mean Acc.	Ensemble Acc.
Baseline (MFCCs+GMMs)	0.725	-
ConvNet (static only)	0.778	0.786
ConvNet (MWFD feature map)	0.761	0.784
ConvNet (MWFD spread, *S1*)	0.814	0.820
ConvNet (MWFD spread, *S2*)	0.820	0.831

Table 2: ASC performance using proposed ConvNet with various input data arrangement methods and aggregation strategies, and ensemble models.

spectrogram input outperformed provided baseline performance. In the case of using MWFD with 4 channel feature map, the performance was slightly lower than the case of using static input only. This result shows that feature map approach is not suitable for MWFD features as expected. On the other hand, putting static and MWFD features as individual examples significantly improved the classification accuracy. We achieved a mean accuracy of 0.814 with $S1$ and the proposed $S2$ aggregation method achieved 0.820.

To increase the performance further, we examined the model ensemble method. We produced ensemble model by combining 3 models generated from the experiment by averaging softmax output probabilities from models. With the model ensemble, MWFD features, and aggregation strategy $S2$, we could increase the accuracy up to 0.831 as shown in Fig.2. Although mean accuracy was 0.831, we could find that the most of the errors occurred from the 'park' scene and it was confused with 'residential area' as illustrated in Fig.3. By listening to actual audio clips of these two scenes, we found that this confusion is mainly due to both scenes are very quiet and mainly include a lot of bird sounds, but confusion happened in one way because residential area also includes car sounds which sometimes captured by our 1s window, sometimes not.

4. EXPERIMENT ON EVALUATION SET

We used same experiment setting with cross-validation for the evaluation set. We used all development dataset audio to train the network, and generated 40 models for ensemble to make the final result stable as possible. Note that the mean and standard deviation values were extracted from the training data (i.e., development dataset) only, and these were used for standardize the evaluation set. As a result, we obtained an accuracy of 0.841 which is close to the result of the development set. This result shows that our system is reasonably stable and not over-fitted to the development set.

5. CONCLUSION

In this paper, we illustrate how we applied ConvNet for ASC task. As a result, proposed ConvNet architecture outperformed given baseline which used MFCCs with GMMs. Also, we proposed MWFD data augmentation and folded mean aggregation to improve the accuracy further. By using MWFD features as individual examples (MWFD spread) and applying folded mean aggregation ($S2$) as well as using average ensemble model, we could obtain classification accuracy of 0.831. We believe that proposed data augmentation and aggregation method is a highly general approach, and planning to examine its usefulness on the other audio/music processing tasks.

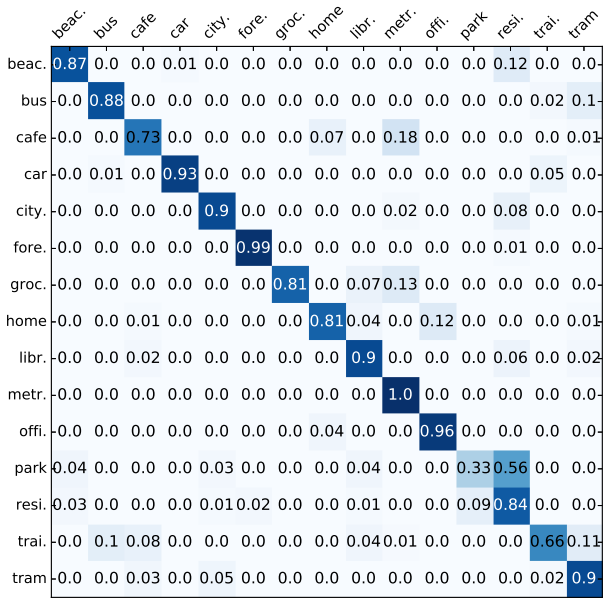

Figure 3: Confusion matrix of the proposed ConvNet with MWFD data augmentation. X axis indicates predicted label and Y axis indicates true label.

6. REFERENCES

[1] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *Signal Processing Magazine, IEEE*, vol. 32, no. 3, pp. 16–34, 2015.

[2] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An ieee aasp challenge," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*, Budapest, Hungary, 2016.

[5] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *ArXiv e-prints*, May 2016.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, 2013, p. 1.

[8] J. Schluter and S. Bock, "Improved musical onset detection with convolutional neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6979–6983.