DOMESTIC ACTIVITIES CLASSIFICATION BASED ON CNN USING SHUFFLING AND MIXING DATA AUGMENTATION

Technical Report

Tadanobu Inoue[1]*, Phongtharin Vinayavekhin*[1]*, Shiqiang Wang*[2]*,*
David Wood[2]*, Nancy Greco*[2]*, Ryuki Tachibana*[1]

[1] IBM Research, Tokyo, Japan, {inouet, pvmilk, ryuki}@jp.ibm.com
[2] IBM Research, Yorktown Heights, NY, USA, {wangshiq, dawood, grecon}@us.ibm.com

ABSTRACT

This technical report describes our proposed design and implementation of the system used for the DCASE 2018 Challenge submission. The work focuses on Task 5 of the challenge, which is about monitoring and classifying domestic activities based on multi-channel acoustics. We propose data augmentation techniques using shuffling and mixing two sounds in a same class to mitigate the unbalanced training dataset. This data augmentation can generate new variations on both the sequence and the density of sound events. The experimental results show that the proposed system achieves an average of 89.95% of macro-averaged F1 score over 4 folds on the development dataset. This is a significant improvement from the baseline result of 84.50%. In the final evaluation for the submission, four proposed classifiers are trained with four folds of training and validation data in the development dataset. Then we ensemble these four models by averaging their predictions.

***Index Terms*—** DCASE 2018 Challenge, Domestic Activities, Multi-channel Acoustics, Deep Learning, Convolutional Neural Network, Data Augmentation

1. INTRODUCTION

In recent years, there is an increasing popularity in installing smart speakers in a home environment due to its voice interface and its capability to interact and activate many home appliances. The low cost of these smart speakers encourages the use of more than one device to cover a greater area of a home. The technology in smart speakers, MEMs array microphones, can be additionally used to monitor sounds, other than human voice dialogue. This work focuses on demonstrating how the speaker capability can be adapted through machine learning to monitor and detect human activities in daily life routine.

The work aims to solve Task 5 of DCASE 2018 Challenge [1]. The challenge is designed using a selected subset of the SINS database [2]. It is posed as a multi-class classification problem. Our proposed system combines two approaches to improve the F1 scores of the classification. First, it augments the input data by shuffling and mixing segments of the sounds. This is based on the characteristic of the input data where events occur discretely in a sound sequence without a temporal relation. This method of data augmentation helps increase the variation in the training samples, and reduces the effect of unbalanced dataset. Second, the proposed system uses a deep learning model as a classifier. Convolutional Neural Networks (CNNs) are widely used for acoustic scene classification tasks [3, 4, 5]. The main characteristic of the proposed network architecture is the use of CNNs on frequency and time axis of log-scaled mel-spectrogram (logmel) representation of the input. First it applies convolutional layers across frequency where the kernel size on the time axis is fixed to one and then it applies a convolutional layer across time where the kernel size on the frequency axis is fixed to one. This allows the network to look for local patterns across frequency bands and also the short connected temporal components which represent sound events in the input data. In addition, the network also maintains the size of the time axis of the logmel until the final pooling layer.

This report starts by describing the data augmentation technique in Section 2. Then, we describe the deep learning based classification model in Section 3. The experimental results on the development dataset of DCASE 2018 Challenge Task 5 are shown in Section 4. Finally, the conclusion is given in Section 5.

2. DATA PREPARATION

2.1. DCASE 2018 Task 5 dataset

The DCASE 2018 Task 5 dataset contains sound data recorded in a living room by individual devices with four microphone arrays at seven undisclosed locations. The dataset is divided into the development dataset and an evaluation dataset. The development dataset contains sounds recorded at one of four undisclosed locations, while the evaluation dataset contains sounds recorded at one of seven locations. Three microphone arrays are not in the development dataset and they are used for the final evaluation score. Each sound data is a 4-channel audio clip with 10 second length and sampled at 16 kHz. Sounds are categorized into nine classes: *absence*, *cooking*, *dishwashing*, *eating*, *other*, *social activity*, *vacuum cleaning*, *watching TV*, and *working*.

We examine the development dataset which is divided into four folds of training and testing data. The data distribution of training and testing set of Fold 1 is shown in Fig. 1.

As shown in Fig. 1, we find that the dataset is unbalanced in the amount of data each class has, which could correspond to a frequency of the activities in real life. The amount of data in the following six classes: *cooking*, *dishwashing*, *eating*, *other*, *social activity*, and *vacuum cleaning*, is extremely small compared to the other three classes: *absence*, *watching TV*, and *working*. Therefore, we decide to increase the amount of sound data for the six classes to mitigate the unbalancing issue using the proposed data augmentation technique.

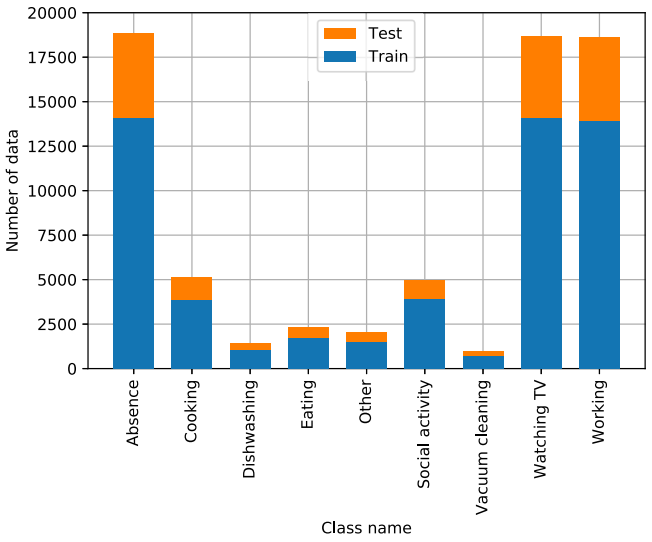

Figure 1: Data distribution of each activity in Fold 1 of the development dataset.

2.2. Data augmentation via shuffling and mixing

We augment sound data based on two assumptions. First, we assume that the acoustic environment does not depend on the order of sound events. For example, let us consider the case of *eating* class as shown in Fig. 2. Some of the sounds categorized as *eating* include sound events made by dish and kitchen utensils as shown in Fig. 2(1). Even when the order of these sound events are swapped, the new sound can still be categorized as *eating*. Therefore, we can generate new data by changing the order of sound events. Second, we assume that mixing two sounds in the same class generates new sound within the same class. This assumption has also been used in previous works [6, 7].

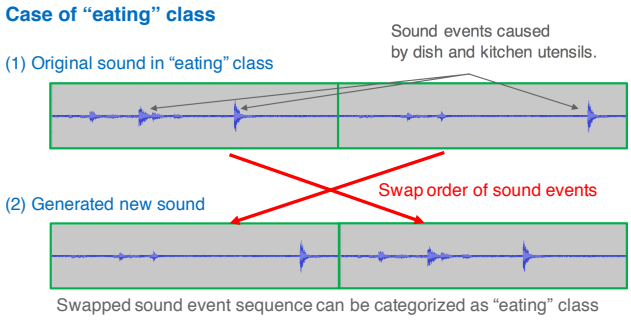

Figure 2: Swapping the order of sound events.

Based on these two assumptions, we proposed a simple but effective data augmentation technique, which is comprised of two steps: (1) shuffling, and (2) mixing two sounds of the same class as shown in Fig. 3.

We divide a 10 second sound sequence into 5 segments. The length of each segment is 2 seconds. As shown in Fig. 3(1), we parepare two ID arrays, sequence ID and sound ID, and shuffle them. Sequence ID represents the sequence of sound segments.

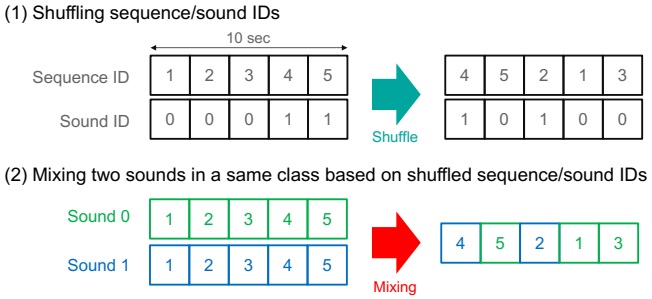

Figure 3: Generating new data based on shuffling and mixing.

For example, when a sequence ID is shuffled from $[1, 2, 3, 4, 5]$ to $[4, 5, 2, 1, 3]$ as shown in Fig. 3(1), it means that the 4th segment of the original sound comes to the first segment of the new sound and the 5th segment comes to the second and the 3rd segment comes to the last. Sound ID represents sound source from two sounds, "Sound 0" or "Sound 1", and how to mix them. For example, when a sound ID is shuffled from $[0, 0, 0, 1, 1]$ to $[1, 0, 1, 0, 0]$ as shown in Fig. 3(1), the first segment of the new sound is picked up from Sound 1 and the second segment is picked from Sound 0 and the 5th segment is picked from Sound 0. As shown in Fig. 3(2), we mix two sounds in a same class based on shuffled sequence/sound IDs.

2.3. Related work

Takahashi *et al.* [6] augment sound data by mixing two sound sources of the same class. Zhang *et al.* [7] mix not only same class sounds, but also different class sounds. They do not augment sound data before the training but instead mixing two sounds on every epoch during the training. Tokozume *et al.* [8] also mix two sounds on every epochs during the training and focus on mixing sounds in different classes.

In these previous works, the mixing kept the order of the sound events and does not generate new variations on the sequence itself. In addtion, their mixing tends to increase the number of sound events by merging two sound events in a linear interpolation manner. On the other hand, our shuffling and mixing approach can have variations on both the order and density of the sound events (the number of the sound events per time period).

3. SYSTEM ARCHITECTURE

In this section, we describe our proposed system architecture. First, preprocessing method for input audio data is explained. The processed audio data is used as an input to the classifier described in Section 3.2. The classifier predicts the class of the input data for each channel independently. The method to merge the results is explained in Section 3.3. Finally, restricted only to the prediction of the evaluation dataset, Section 3.4 describes the ensemble method which is used to merge the results of multiple classifiers which are trained from all folds of the development dataset.

3.1. Audio preprocessing

Each 10 second audio clip is preprocessed in the way similar to the baseline system of the DCASE 2018 Challenge:

- We do not apply any normalization or standardization.

- The frame size for short-time Fourier transform is 64 ms with a hop size of 20 ms. This is longer than the frame size of the baseline system, 40 ms.
- 40-bin logmel that transforms raw audio data into a 40×501 matrix.
- Data from each channel is treated independently.

3.2. Network architecture

The proposed deep neural network architecture stack together the following neural network layers: CNNs layer, batch normalization (BN) [9], relu activation, max and global max pooling, dropout, fully connected layer, and softmax layer. First the proposed network applies convolutional layers across frequency where the kernel size on the time axis is fixed to one and then it applies a convolutional layer across time where the kernel size on the frequency axis is fixed to one. The max pooling is performed only on the frequency axis, while the size of temporal axis is maintained to be the same until the global max pooling layer. This network structure is inspired by the model used on a speech recognition task [10]. The complete network architecture and parameters are shown in Table. 1. The last layer of the network is a softmax layer, hence the prediction is a probablity of the input sound being in each class.

Table 1: Proposed network architecture.

Layer	Output size
Input	$40 \times 501 \times 1$
Conv(7×1, 64) + BN + ReLU	$40 \times 501 \times 64$
Max pooling(4×1) + Dropout(0.2)	$10 \times 501 \times 64$
Conv(10×1, 128) + BN + ReLU	$1 \times 501 \times 128$
Conv(1×7, 256) + BN + ReLU	$1 \times 501 \times 256$
Global max pooling + Dropout(0.5)	256
Dense	128
Softmax output	9

3.3. Fusing multiple channel audio data

In the dataset, one audio clip is captured from a microphone array and contains 4-channel sounds. Each channel sound is preprocessed and pass through the classifier independently. This results in four predictions for one audio clip. An average of these four softmax predictions is calculated as a final probability prediction for each sound clip.

3.4. Ensemble method for final submission

In the final evaluation for submission, four different classifiers are obtained by training the same network architecture with four folds of training and validation data. We reuse original definition of Folds $1-4$ for this purpose. The training set is the same during the development and the final evaluation for each fold. However, the original testing sets during the development are used as the validation sets for training the model for the final evaluation as shown in Fig. 4.

This results in four different deep neural network models for the evaluation dataset. We ensemble these four models by averaging the softmax outputs from models with an equal weight.

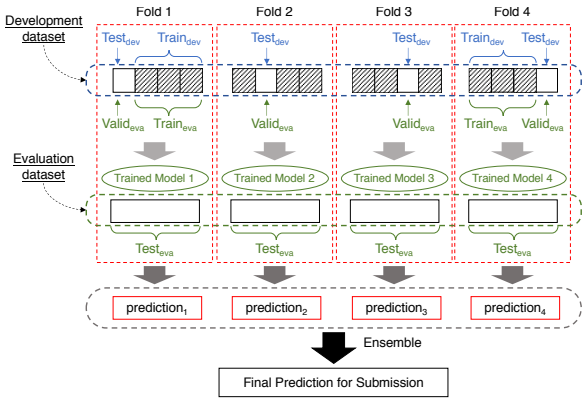

Figure 4: A final submission combines results from four different classifier models trained with four different training/validation sets.

4. EXPERIMENTAL RESULTS

4.1. Classification on development dataset

The experments are carried out using the 4-fold cross validation setting in the development dataset provided by the organizer of the DCASE 2018 Challenge.

We select 30% of the training data as validation data by making sure that all segments from the same session must be either in the training or validation data. This is the same as how it is done in the baseline system [1]. The proposed data augmentation appraoch is then applied to the training data of the six classes that have less amount of data, as discussed in Section 2.1. Fig. 5 shows data augmentation results on Fold 1 of the development dataset. The red dotted line in Fig. 5(2) shows the increased dataset by shuffling and mixing.

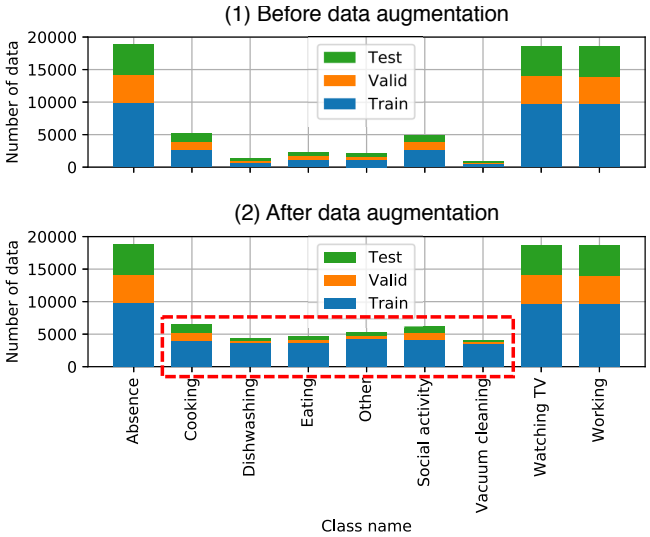

Figure 5: Data augmentation in fold1 of the development dataset.

We utilize ADAM optimizer [11] with an initial learning rate of 0.0001 and a batch size of 256 samples. We train the classifier

[1] dcase_util/datasets/datasets.py: validation_files_balanced

for 500 epochs and choose network weights which result in the best accuracy on the validation data as the final network weights. We examine the following three configurations:

(1) Proposed CNN without data augmentation to check the effect of CNN design.

(2) Baseline CNN with proposed data augmentation to check the effect of data augmentation.

(3) Proposed CNN with proposed data augmentation to check the total effect (proposed system).

Fig. 6 shows the F1 scores of overall and each class by the baseline system and the above three configurations.

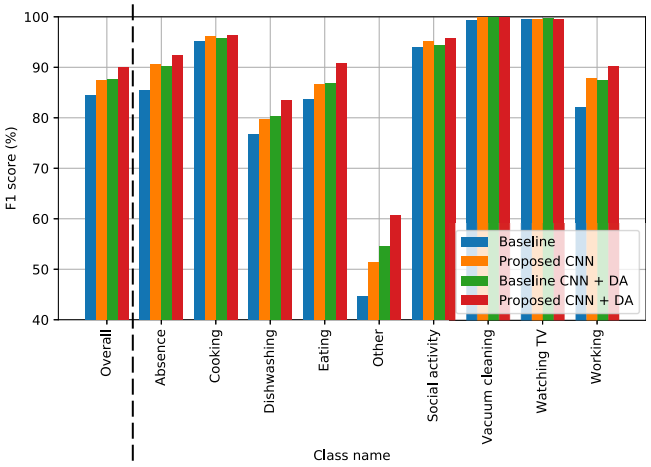

Figure 6: F1 scores on each class.

As shown in Fig. 6, we can see that the proposed network architecture and data augmentation approach improve the classification performance respectively and the combination of them gives the best performance. The overall F1 score by the proposed system is 89.95%, while the overall F1 score by the baseline is 84.50%. The proposed system improves F1 scores in all classes, especially the F1 score of *other* class.

4.2. Classification on evaluation dataset

As described in Fig. 4, we train four models using the proposed approach on the four different folds of the development dataset and predict the evaluation dataset by all trained classifiers. The final prediction result is generated by averaging these four prediction results. We train models and predict the evaluation dataset twice and then submit these two results for the final submission.

5. CONCLUSIONS

In this technical report, we illustrate how we apply deep convolutional neural network with data augmentation techniques for DCASE 2018 Task 5 for monitoring domestic activities. We propose a data augmentation technique that shuffles and mixes two sounds in the same class to mitigate the unbalanced training dataset. This data augmentation can generate new variations on both the sequence and the density of sound events. On the development dataset, the proposed system achieves 89.95% overall F1 score which is significantly improved from the baseline performance, 84.50%. In the final evaluation for the submission, four proposed classifiers are trained with four folds of training and validation data in the development dataset. Then we ensemble these four models by averaging their predictions.

6. REFERENCES

[1] G. Dekkers, L. Vuegen, T. van Waterschoot, B. Vanrumste, and P. Karsmakers, "DCASE 2018 Challenge - Task 5: Monitoring of domestic activities based on multi-channel acoustics," KU Leuven, Tech. Rep., July 2018.

[2] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. a. Karsmakers, "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," in *DCASE 2017*, November 2017.

[3] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *MLSP 2015*, 2015.

[4] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," in *IEEE Signal Processing Letters*, vol. 24, 2017, pp. 279–283.

[5] Y. Han, J. Park, and K. Lee, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," in *DCASE 2017*, 2017.

[6] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, "Deep convolutional neural networks and data augmentation for acoustic event recognition," in *INTERSPEECH 2016*, September 2016.

[7] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR 2018*, 2018.

[8] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," in *ICLR 2018*, 2018.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML 2015*, 2015.

[10] "Deep learning model for speech recognition," https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/discussion/47715, accessed: 2018-07-30.

[11] D. P. Kingma and J. Lei Ba, "Adam: a method for stochastic optimization," in *ICLR 2015*, 2015.