DOMESTIC AUDIO TAGGING WITH CONVOLUTIONAL NEURAL NETWORKS

Emre Çakır, Toni Heittola and Tuomas Virtanen

Tampere Universiy of Technology, 33101 Tampere, Finland

ABSTRACT

In this paper, the method used in our submission for DCASE2016 challenge task 4 (domestic audio tagging) is described. The use of convolutional neural networks (CNN) to label the audio signals recorded in a domestic (home) environment is investigated. A relative 23.8% improvement over the Gaussian mixture model (GMM) baseline method is observed over the development dataset for the challenge.

Index Terms— Audio tagging, sound event classification, convolutional neural networks

1. INTRODUCTION

The aim of domestic audio tagging is to tag/label an audio recording from a home environment with one (or more) of the pre-determined sound sources present in the recording. Smartphones and similar electronic gadgets with sound recording capabilities significantly increased the amount of sound recordings from home environments. Automatic labeling of these recordings can be utilized in many application areas, including lifelogging [1] and health activity monitoring [2].

2. METHOD

The method can be grouped into two stages: *sound representation* and *classification*. In sound representation stage, the audio waveform is transformed into a sequence of spectral domain feature vectors extracted from short time frames. In classification stage, a deep convolutional neural network (CNN) is trained to obtain source presence probabilities for each frame. The source presence probabilities for each recording is calculated by taking the average of the probabilities over the short time frames of the recording.

2.1. Sound representation

The audio waveform is first converted into mono by simply averaging over two channels. Then, it is divided into 40 ms frames with 50% overlap and then multiplied with a Hamming window. For each frame, magnitude response is calculated using short time Fourier transform (STFT) with 1024 points. Then, total energies in 40 mel bands are calculated over the magnitude response to be used as sound features $\mathbf{x}_t$ for each frame t. Finally, each feature is normalized for zero mean and unit standard deviation. In order to make use of the temporal structure of the audio recordings, the input to the classifier is represented as blocks of frames $\mathbf{X} \in \mathbb{R}^{40\times32}$, where a block of 32 frames corresponds to a context window of 640 miliseconds.

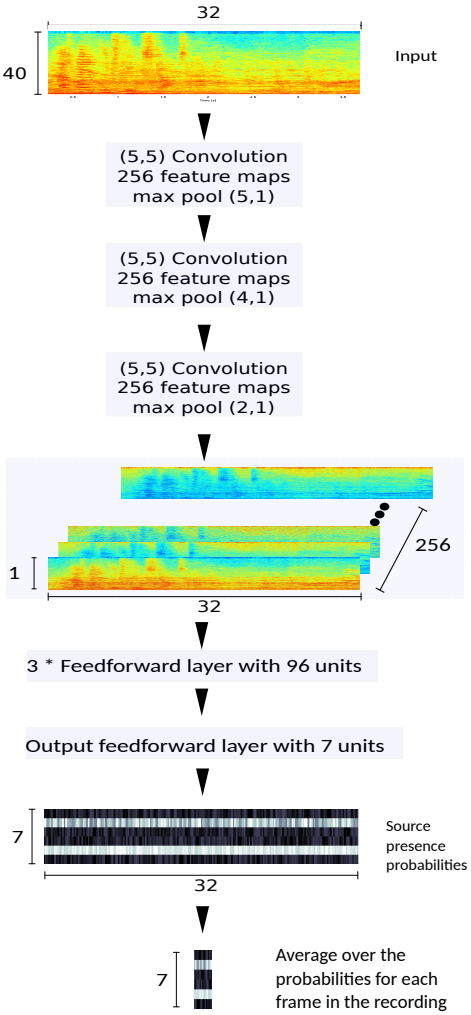

Figure 1: Overview of the utilized CNN method.

2.2. Classification

The overview of the method is illustrated in Figure 1. In order to obtain the source presence probabilities for each frame, a deep CNN with three convolutional layers (with frequency max-pooling), three feedforward layers as hidden layers and one feedforward layer as output layer is trained. Each convolutional layer has 256 feature maps with shape $(5, 5)$, therefore convolution applied both in time

Table 1: Equal error rate (EER) for baseline and CNN methods.

Sound source	Fold 1 CNN	Fold 2 CNN	Fold 3 CNN	Fold 4 CNN	Fold 5 CNN	Average Baseline	Average CNN
adult female speech (f)	0.27	0.22	0.15	0.33	0.25	0.29	**0.25**
adult male speech (m)	0.18	0.12	0.11	0.18	0.19	0.30	**0.15**
broadband noise (b)	0.00	0.00	0.00	0.00	0.33	0.09	**0.07**
child speech (c)	0.24	0.14	0.23	0.22	0.24	**0.20**	0.21
other (o)	0.27	0.26	0.12	0.31	0.33	0.29	**0.26**
percussive sound (p)	0.31	0.10	0.19	0.24	0.23	0.25	**0.21**
video game/tv (v)	0.15	0.03	0.03	0.01	0.05	0.07	**0.05**
Mean EER	0.20	0.13	0.12	0.18	0.23	0.2129	**0.1714**

and frequency domain. The outputs of each convolutional and feedforward hidden layers are followed by batch normalization [3] and rectified linear unit (ReLU) activation function. Convolutional layer outputs are passed through max pooling in frequency domain. The initial 40 input features per frame is downsampled in to a single feature per feature map over three convolutional layers with pool sizes 5, 4 and 2. Pooling is not applied in time domain in order to obtain source presence probability for each frame. Therefore the output of the final convolutional layer is $\mathbf{H} \in \mathbb{R}^{256\times32}$. Dropout [4] with probability 0.25 is applied for each convolutional and feedforward hidden layer. Three consecutive feedforward hidden layers with 96 hidden units is placed after the convolutional layers. At each feedforward layer, same weights and biases are applied for each of the 32 frame outputs of $\mathbf{H}$. The output feedforward layer has logistic sigmoid activation function and the outputs from the output layer are treated as source presence probabilities in each frame. The target output for each frame t is a binary vector $\mathbf{y}_t \in \mathbb{R}^K$ where K is the number of pre-determined sound sources ($K = 7$ for the challenge). If the source K is present in a recording, the target output $\mathbf{y}_t(k)$ for each frame from the recording will be set as 1 and 0 vice versa. The estimated source presence probabilities $\hat{\mathbf{y}} \in \mathbb{R}^K$ for each recording is determined as the average of the source presence probabilities of $\hat{\mathbf{Y}} \in \mathbb{R}^{K\times T}$ the recording, where T is the number of frames in the recording. The network is trained with stochastic gradient descent with cross entropy as loss function and Adam [5] as the learning rate schedule optimizer. Keras deep learning package is used in this work [6].

3. EVALUATION

3.1. Dataset and Evaluation Metric

The utilized method is evaluated on CHIME-HOME development dataset [7], which is the official dataset for DCASE2016 challenge task 4 [8]. The dataset consists of 4378 chunks of audio recordings from home environments. Seven sound sources have been determined for this dataset as child speech (c), adult male speech (m), adult female speech (f), video game / TV (v), percussive sounds (p), broadband noise (b) and other identifiable sounds (o). Each chunk is annotated with one (or more) of these sound sources. The official evaluation metric for the challenge is average equal error rate (EER) for five-fold cross-validation.

3.2. Results

The evaluation results for the CNN method over DCASE2016 challenge task 4 development dataset is presented in Table 1. For comparison, the results for the baseline method [7] is also presented in the same table. For sound representation, baseline method uses 14 mel frequency cepstral coefficients (MFCC) extracted from 20 ms frames with %50 overlap. As the classifier, baseline method uses Gaussian mixture modeling (GMM) with 8 Gaussians to model each sound source.

The utilized CNN method achieves 18.4% relative improvement over the baseline method. Considering the results per sound source, baseline method and CNN achieve quite similar performance, with exception of significantly better performance of CNN for *adult male speech*.

4. DISCUSSIONS

The recordings for the challenge have two important properties that create difficulty for tagging task. First one is the environmental noise due to the fact that the recordings are obtained from real-life environments. This makes it necessary for the proposed tagging method to be noise robust. CNNs are able to extract higher level features that are invariant to local spectral and temporal variations. This may explain their improved performance especially on both male and female speech, where the frequency response can exhibit small variations between humans. Second property is that multiple sound sources can be present in a single recording at the same time. Therefore, the classifier should be able to model and recognize multiple sound sources simultaneously. The multiple feature maps of the convolutional layers and different subsets of hidden units of feedforward layers can help modeling different sound sources simultaneously, which is another advantage of the utilized CNN method.

5. REFERENCES

[1] M. Shah, B. Mears, C. Chakrabarti, and A. Spanias, "Lifelogging: Archival and retrieval of continuously recorded audio using wearable devices," in *IEEE International Conference on Emerging Signal Processing Applications (ESPA)*. IEEE, 2012, pp. 99–102.

[2] S. Goetze, J. Schroder, S. Gerlach, D. Hollosi, J.-E. Appell, and F. Wallhoff, "Acoustic monitoring and localization for social care," *Journal of Computing Science and Engineering*, vol. 6, no. 1, pp. 40–50, 2012.

[3] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[4] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[5] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[6] F. Chollet, "Keras," https://github.com/fchollet/keras, 2016.

[7] P. Foster, S. Sigtia, S. Krstulovic, J. Barker, and M. D. Plumbley, "Chime-home: A dataset for sound source recognition in a domestic environment," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2015, pp. 1–5.

[8] "DCASE2016 challenge task 4," http://www.cs.tut.fi/sgn/arg/dcase2016/task-audio-tagging, accessed: 2016-07-01.