Convolutional Neural Network based Audio Event Classification

Minkyu Lim[1], Donghyun Lee[1], Hosung Park[1], Yoseb Kang[1], Junseok Oh[1], Jeong-Sik Park[2], Gil-Jin Jang[3], and Ji-Hwan Kim[1]

[1] Dept. of Computer Science and Engineering, Sogang University
Seoul, Republic of Korea
[e-mail: {lmkhi, redizard, hosungpark, kjoe, ohjs, kimjihwan}@sogang.ac.kr]
[2] Dept. of English Linguistics & Language Technology, Hankuk University of Foreign Studies
Seoul, Republic of Korea
[e-mail: parkjs@hufs.ac.kr]
[3] School of Electronics Engineering, Kyungpook National University
Daegu, Republic of Korea
[e-mail: gjang@knu.ac.kr]
*Corresponding author: Ji-Hwan Kim

Received May 29, 2017; revised August 29, 2017; accepted February 13, 2018;
published June 30, 2018

Abstract

This paper proposes an audio event classification method based on convolutional neural networks (CNNs). CNN has great advantages of distinguishing complex shapes of image. Proposed system uses the features of audio sound as an input image of CNN. Mel scale filter bank features are extracted from each frame, then the features are concatenated over 40 consecutive frames and as a result, the concatenated frames are regarded as an input image. The output layer of CNN generates probabilities of audio event (e.g. dogs bark, siren, forest). The event probabilities for all images in an audio segment are accumulated, then the audio event having the highest accumulated probability is determined to be the classification result. This proposed method classified thirty audio events with the accuracy of 81.5% for the UrbanSound8K, BBC Sound FX, DCASE2016, and FREESOUND dataset.

Keywords: Audio event classification, Convolutional neural networks, Deep learning

This work was supported by the ICT R&D program of MSIP/IITP. [2015-0-00225, Development of Media Application Framework based on Multi-modality which enables Personal Media Reconstruction

http://doi.org/10.3837/tiis.2018.06.017 **ISSN : 1976-7277**

1. Introduction

Daily media use has gradually shifted from mass media (eg. TV shows and movies) to personal media such as user-created contents [1]. Until recently, most analysis of multi media depended on metadata. In this reason, leading global software companies actively conduct research of automatic segmentation, tagging, context classification through the analysis of video content. To analyze the meaning of video content, the classification of sound events included in the videos is essential. Regarding of conventional sound event classification, most studies have focused on extracting various features, such as spectral flux, zero crossing rate (ZCR), and band periodicity, including verifying their performances or on Gaussian mixture models (GMM) and rule-based classifiers [2]-[4]. In those studies, however, the number of classes in sound events was limited to music, voice, and other sounds.

Deep neural networks (DNNs) have attracted attention as a technology that shows better performance than conventional methods in various machine-learning fields. DNN is an artificial neural network composed of two or more hidden layers. It can distinguish more complex and nonlinear learning boundaries than artificial neural networks with a single hidden layer, exhibiting better performance in classification problems. A large amount of data, however, is necessary to estimate the numerous parameters of DNN, and considerable computation is required. DNNs have been applied to various applications because of recent advances in big data and hardware technologies.

In particular, convolutional neural networks (CNNs) have been applied to image classification, showing remarkable performance improvements. CNN has great advantages of distinguishing complex shapes of image. Proposed system uses the features of audio sound as an input image of CNN. Mel scale filter bank features are extracted from each frame, then the features are concatenated over 40 consecutive frames and as a result, the concatenated frames are regarded as an input image. The output layer of CNN generates probabilities of audio event (e.g. dogs bark, siren, forest). The event probabilities for all images in an audio segment are accumulated, then the audio event having the highest accumulated probability is determined to be the classification result.

The structure of this paper is as follows. In Section 2, previous studies on audio event classification are introduced. In Section 3, an audio event classification method using CNNs is described. In Section 4, contents of the audio-event selection, data refinement, and experiment for efficient audio event classification from personal media are described. Our conclusion is provided in Section 5.

2. Related Work

Recently, there was a study that applied deep belief networks (DBNs) to musician- and music genre-classification problems [5]. The DBN is a method of sequentially providing learning to hidden layers using various learning data without answer scripts, and then using a small amount of data with answer scripts to finally provide learning to output layers. The hidden layer learning of DBNs stacks the hidden layers that went through unsupervised learning using a restricted boltzmann machine (RBM). Supervised learning is finally performed for output layers only using a small amount of learning data with answer scripts. According to the experimental results, an approximately 73 % classification performance was observed in the five music genre classification experiments, and an approximately 80 % classification performance was observed in the four musician classification experiments. This study shows the advantage of using a large amount of training data without answer scripts. There is a limitation, however, that the number of classes is small in the classification experiments.

Recently, there was a study that analyzed the difference between the existing support vector machine (SVM) and the DNN [6]. In this study, the SVM and DNN were classified as shallow architecture and deep architecture, respectively. In the case of an SVM, it reduces the dimensions through the kernel and draws lines to classify the class distribution and can be seen as an artificial neural network with a single hidden layer. On the other hand, a DNN having multiple hidden layers can create more complex decision boundaries than SVMs or artificial neural networks with a single hidden layer because it stacks nonlinear boundaries. A large amount of data, however, is required to learn DNN parameters.

A study on hidden markov model (HMM)-SVM-based audio event classifiers was recently conducted [7]. Fifteen audio event classifiers was trained using various feature vector combinations including the mel frequency cepstral coefficient (MFCC), perceptual linear prediction (PLP), and zero crossing rate (ZCR). It was found that recognition performance could be different depending on the feature vector combinations. Events were detected by generating answer scripts directly from sound events in talk shows, news, movies, and documentaries. Experimental results showed that the use of PLP feature vectors exhibited the highest performance.

There was also a recent study involving an experiment on classifying five types of events using a DBN from the sound information of a sports broadcast video [8]. The five events were crowd sound, commentary, crowd sound+commentary, excited commentary, and silence. The performance of trained DBN model was compared with that of SVM model. Experimental results show that the performance of the SVM was slightly higher than that of the DNN. It was determined that this was caused by machine learning with a deficit of training data.

A study was recently conducted on sound event sequence classification in which multiple events existed in one sample [9]. Previous studies of sound event classification, it was assumed that only one event existed in a sample. However, as there are cases with various events, the event sequence was classified. After modeling each sound event by GMM, a 3-state hidden markov model (HMM) was used to classify the sound event sequence. While this model exhibits high performance when sound events appear in a sequence, it is difficult to collect training data with answer scripts.

There was a study on distinguishing four different events: traffic, music, crowd, and applause using an RBM [10]. In this study, RBM was trained so that the hidden layer could generate an output feature vector for an input feature vector; then the feed forward neural network (FFNN) was configured for the highest output layer so that the sound event could be produced. DNNs using RBM were evaluated together with SVM and GMM, and it was found that RBM had better classification performance than GMM and SVM. This study also had a limitation that the number of sound events was small.

Competitions and public challenges related to audio information analysis focus primarily on voice and music. CHiME dealt with keyword recognition in noisy environments [11]; keyword recognition performance in various noisy environments was comparatively evaluated, with the result that the measured performance of human recognition and that of the classifier with the best performance showed a difference in recognition rate of about 5%. MIREX assigns tasks related to music information retrieval (MIR) such as music beat tracking, chord estimation, melody extraction, and genre classification [12]. The CLEAR assignment evaluates systems that recognize human behavior, reactions, and information from the surrounding environment using multimodal video/audio information. Audio environment analysis is included, and an HMM-based system that performs recognition through sound in nine environments (such as the airport, street, or bus in CMU) showed an error rate of approximately 15% [13]. The TRECVID challenge mainly analyzes videos; it has three specific assignments: semantic indexing (SIN); multimedia event detection (MED), which includes audio event analysis; and localization (LOC) [14]. The SiSEC challenge focuses on source classification issues in audio with mixed music and voice [15].

A study that classified audio events by frame, based on feed-forward neural net (FFNN), was also conducted [16]. A method to classify sound events in every frame was proposed. The FFNN consists of one input layer, one output layer, two hidden layers. Each hidden layer is composed of 2,000 neurons, and each neuron has weight and bias as parameters. The feature vector generated from the audio feature extractor becomes the input vector in the FFNN, which makes up the audio event classifier. The values of the output layer corresponding to the input vector represent the occurrence probability of each class. When one sample input was classified, the probabilities of each event for all the frames were added, and the event with the highest probability was determined as the classification result. The number of audio events was 10, and the audio event classification accuracy of sample units showed more than 70 % performance showing better performance than SVM.

3. CNN Based Audio Event Classification

The proposed CNN based audio event classifier consists of two modules: an audio feature extractor and a CNN-based frame level audio event classifier. **Fig. 1** shows the overall audio event classification process. In audio feature extractor, audio signal is converted to raw pcm format, and then mel-scale filter bank features are extracted. Audio event classifier is composed of convolutional layer, pooling layer, and fully connected multi-layer perceptron (MLP). Section 3.1 describes details of the audio feature extractor and Section 3.2 describes the CNN-based frame level audio event classifier. Audio event detection method from frame level classification result is described in Section 3.3.

3.1 Audio Feature Extractor

The audio feature extractor creates a feature vector sequence for a given input signal, and the vector sequence is used in the input layer of the CNN. The input audio signal is converted in raw pcm format with a sample rate of 16,000 samples/sec and two bytes per sample. The hamming window in a 20 ms size moves by 10 ms for the input signal and performs short-time Fourier transform (STFT) for each window. A triangular bin that increases by mel-scale is applied to all windows and each frequency energy is multiplied by weights to extract one feature value for each triangular bin. A mel-scale filter bank (FBANK) feature vector is created by forming a vector using these feature values. 40-dimensional FBANK features are extracted from each frame, then the features are concatenated over 40 consecutive frames. As a result, the concatenated frames are regarded as an input image.

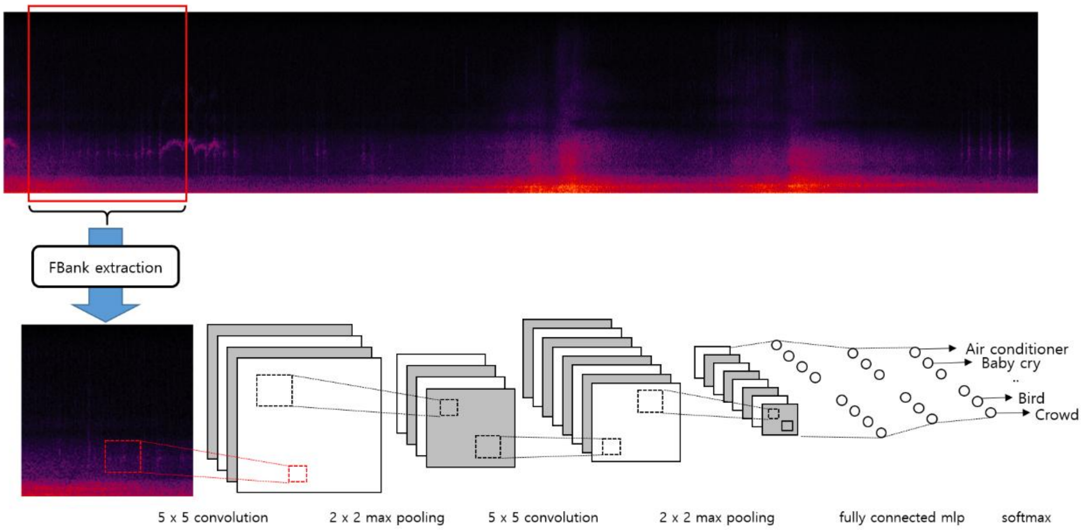


Fig. 1. Structure of CNN based audio event classifier

3.2 Audio Event Classifier

CNN based audio event classifier has three main components: convolutional layer, pooling layer, and fully connected layer. Convolutional layer computes the output of neurons that are connected to local patches in the input. This local patches, generating a dot product between weights and sub-region of the image, are shifted from top-left to bottom-right of the image. The role of the convolutional layer is known as to detect local conjunctions of features from the previous layer [22]. Pooling layer performs a non-linear downsampling operation along the spatial dimensions of the previous layer, resulting in dimension reduction and merging sematically similar features into one [22]. Finally, the fully connected MLP layer is used to classify an audio event of input image. The event probabilities for all images in an audio segment are accumulated, and then the audio event having the highest accumulated probability is determined to be the classification result.

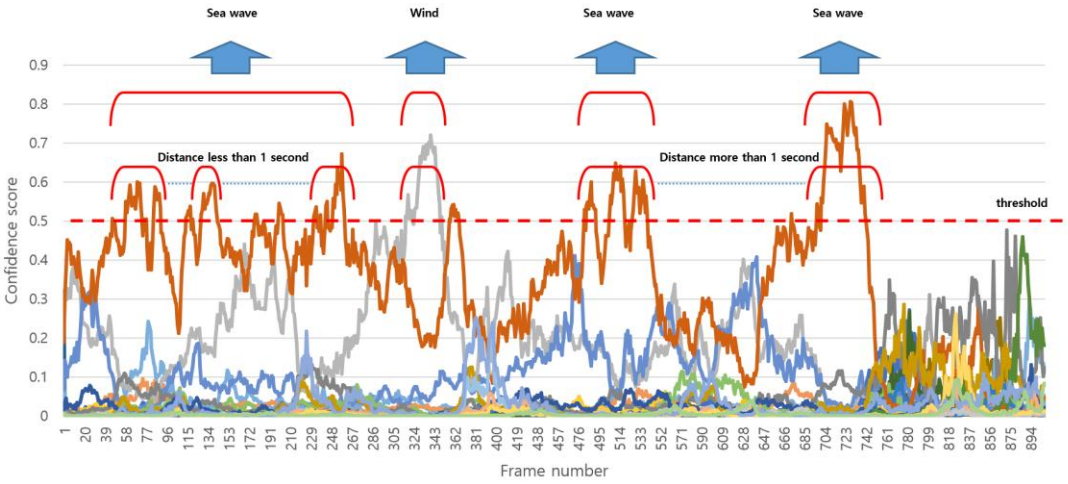


Fig. 2. Example of event detection for sea wave sound

3.3 Audio Event Detection

The segment detection from the frame-level classification result is performed as shown in Fig. 2. The occurrence probabilities of all the audio objects are calculated per frame through CNN forward propagation. If the occurrence probability in more than five consecutive frames exceeds the learned threshold, the corresponding audio event is derived from the first segment detection. If the first detected segments with the same audio event are adjacent to each other within one second, the corresponding two segments are combined into one. Finally, the audio object name, start time, and end time are derived as audio object detection results.

4. Experiments

4.1 Audio Event List

To conduct audio event analysis in as many segments as possible, the output class of CNNs should match the sounds from video as much as possible. To analyze this, total 11 hours of youtube videos were collected and audio events are tagged by annotators. Afterwards, 30 events were selected in the order of highest occurrence. **Table 1** gives the selected audio event list.

Table 1. Audio event list

air conditioner	baby cry	bird
boiling	cafe	car horn
car roads	children playing	city center
cow	crowd	dog bark
drilling	engine idling	forest
grocery store	gun shot	horse
jack hammer	metro station	office
park	rain	river
sea wave	ship	siren
street music	train	wind

4.2 Dataset

The data corresponding to the list of the audio events in **Table 1** were collected from four datasets: UrbanSound8K, BBC Sound FX, DCASE2016, FREESOUND.

4.2.1 UrbanSound8K

UrbanSound8K refers to audio event data collected and distributed by New York University. It is a collection of sounds that can occur in everyday urban life [17]. One audio sample lasts less than four seconds, and a total of 8,732 files are provided. It has 10 audio event classes which are subset of **Table 1**, and has a total length of nine hours. All audio samples are refined well so that additional data preparation is not required.

4.2.2 BBC Sound FX

BBC Sound FX dataset is a collection of sound effects used in movies and TV broadcasts, and consists of 40 CDs [18]. As each track is made up of a large theme, separate analysis and tagging are required. In this study, manual tagging was performed on audio events that can be classified for all the tracks. Tagging results show that a total of 160 audio events exist. Among them, 10 audio events, which belong to the events selected in **Table 1** and contain enough quantity to be used as training data, are selected, and the corresponding tagging results are selected as training data.

4.2.3 DCASE 2016

DCASE 2016 is a challenge that performs audio classification evaluation on four tasks [19]. Among them, task1 contains a considerable number of events selected in **Table 1** as the contents for audio scene classification. Like UrbanSound8K, it is a refined dataset without a need for separate tagging and is used as training data.

4.2.4 FREESOUND

FREESOUND is a cloud database where any user in the cloud shares sounds with specific keywords. As duplicate events exist when classes are constructed from the preceding three datasets, data are collected from freesound.org to supplement the duplicate events. As the collected data may contain errors, manual tagging was performed for segments with actual sounds.

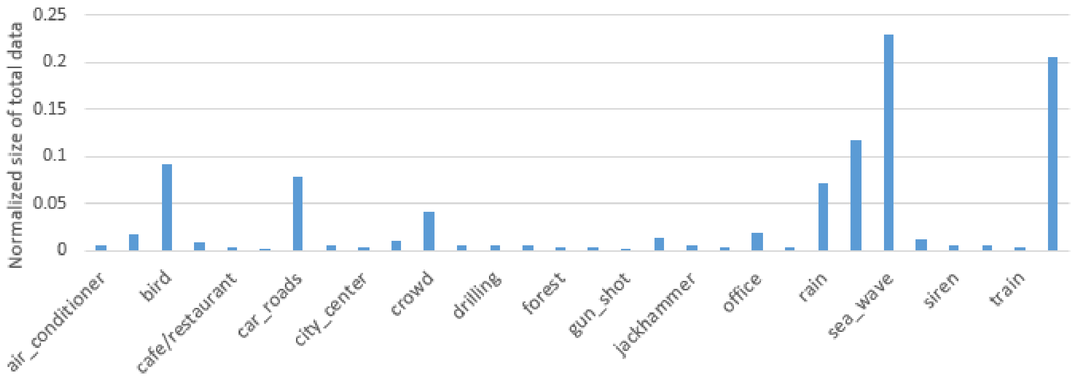


Fig. 3. Data size distribution according to each audio event class

As the datasets are collected from different domains, the total number of segments and playback time are different for each audio event. In particular, as the playback time is much different for each event, additional processing is required for extracting features. **Fig. 3** shows the proportion of the amount of learning data for each event.

Audio events that occupy large quantities are those corresponding to environmental sounds, such as rain, river, wind, and sea waves. These sounds have a very long playback time per audio segment, exhibit little sound change, and show the characteristics that similar sounds are repeated. Owing to the nature of CNN training, when most of the images are similar, even if there are a large amount of data, it is disadvantageous in terms of generalization. Therefore, the frame shift size for each event is determined in proportion to the amount of learning data according to event, and the extracted training images are eventually generated in the same amount for each event.

4.4 Experiment Results

The HTK [20] toolkit is used for FBANK feature extraction, and DNN/CNN model learning is implemented using tensorflow [21]. One-tenth of the training data is used as validation data. Test data consists of a total of 600 audio segments by selecting 20 segments-per-event from all the datasets. All the data are converted to 16 kHz sampling and 16 bit-mono. To evaluate the performances of the CNN-based model and the baseline DNN-based model, 7 models were constructed as shown in **Table 2**.

Table 2. DNN / CNN model description

Model name	Model description
DNN1	4-hidden layer, 1,000 neurons per layer
DNN2	4-hidden layer, 2,000 neurons per layer
DNN3	4-hidden layer, 3,000 neurons per layer
CNN1	5x5 conv - 2x2 max pool - 5x5 conv - 2x2 max pool - 1024 fully connected
CNN2	5x5 conv - 2x2 max pool - 4x4 conv - 2x2 max pool - 3x3 conv - 2x2 max pool -1024 fully connected
CNN3	5x5 conv - 5x5 conv - 2x2 max pool - 4x4 conv - 4x4 conv - 2x2 max pool - 3x3 conv - 2x2 max pool - 2048 fully connected - 1024 fully connected
CNN4	5x5 conv - 5x5 conv - 2x2 max pool - 4x4 conv - 4x4 conv - 2x2 max pool - 3x3 conv - 2x2 max pool - 3x3 conv - 2x2 max pool - 2048 fully connected - 1024 fully connected

Fig. 4 shows the frame level validation accuracy for each epoch. Experimental results show that the performance of CNNs is much higher than that of the baseline DNN-based audio classification model. **Fig. 5** shows loss value according to training models.

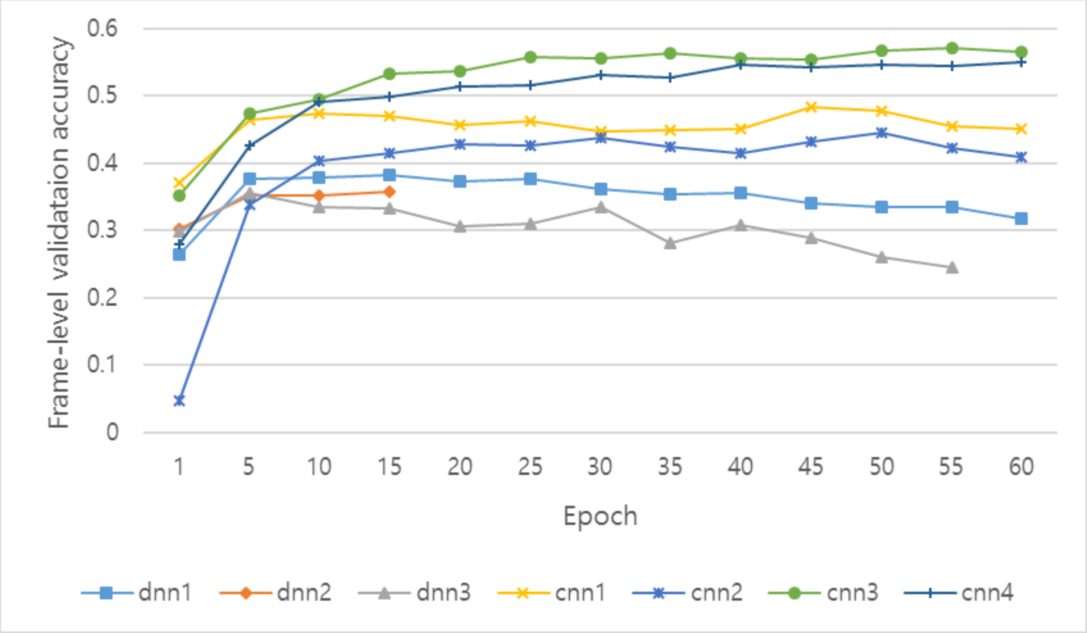


Fig. 4. Frame level validation accuracy

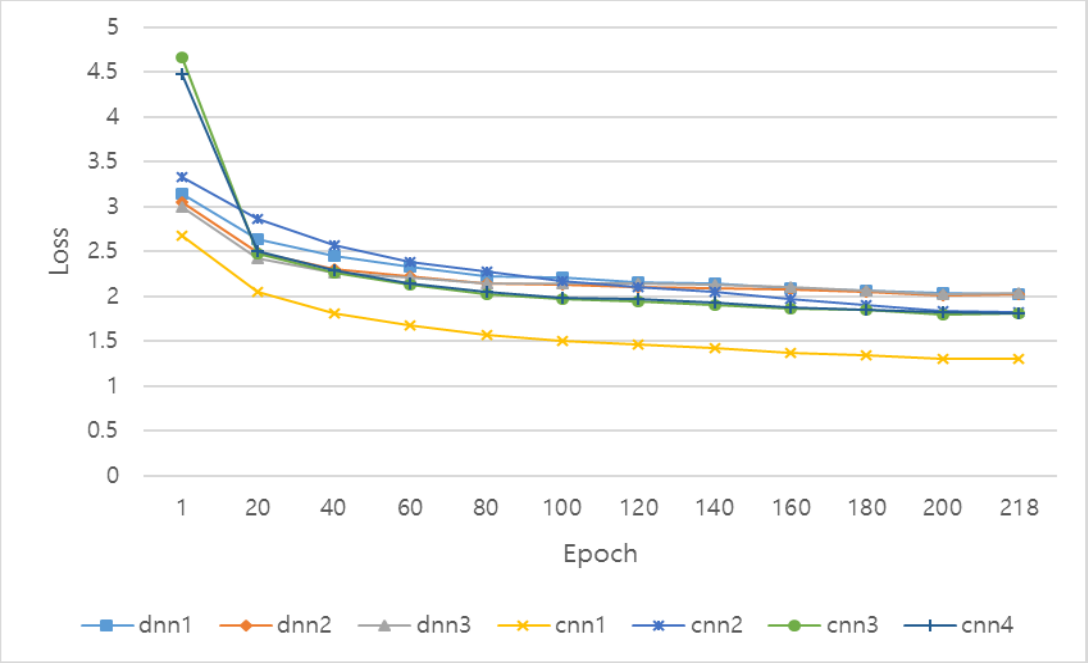


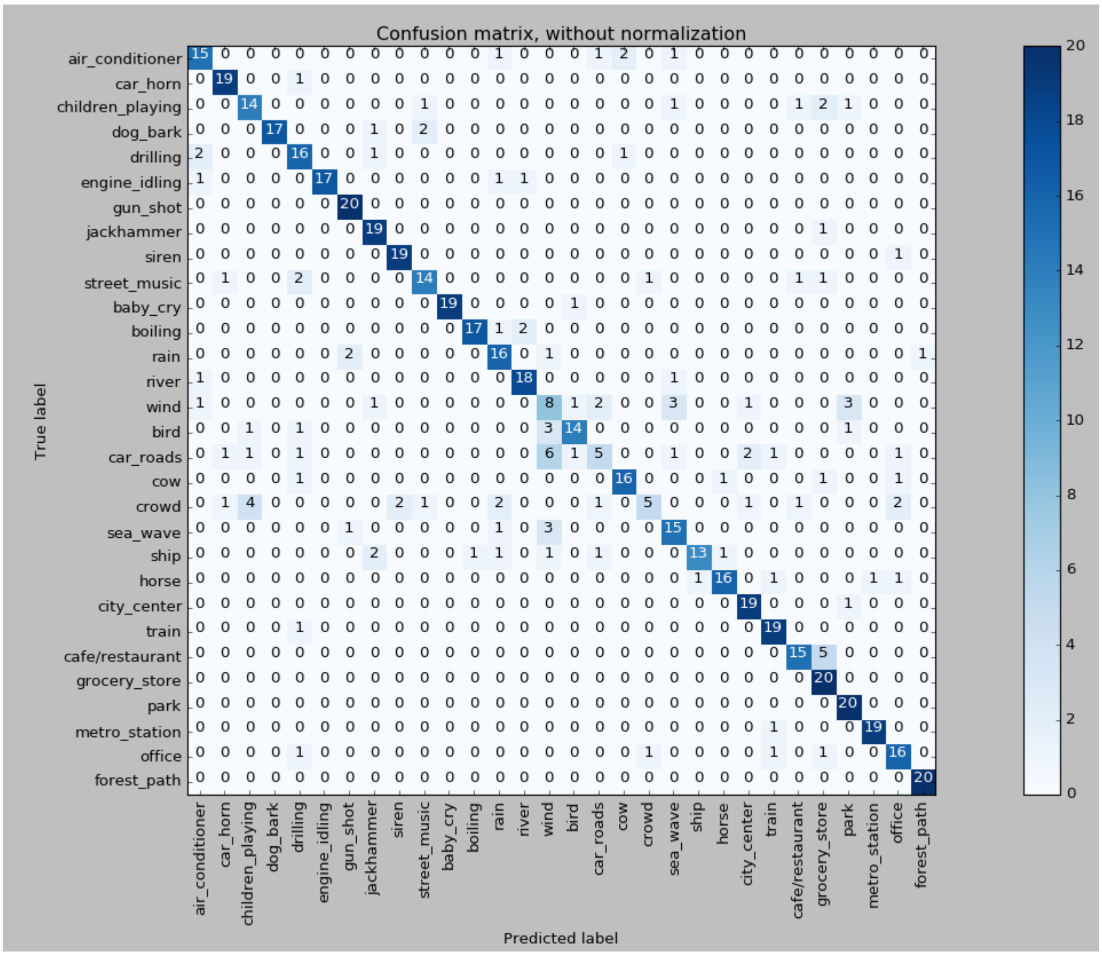
Fig. 5. Loss value according to training models

The parameters of the epoch with the highest validation accuracy are determined as the final parameters for each model. **Table 3** gives the test results. Frame level accuracy indicates the frame level accuracy for a total of 600 test data. Segment level accuracy indicates the segment level accuracy for 600 segments. Segment level classification tests were conducted by using two decision making methods: probability accumulation method and voting method. In the probability accumulation method, the event probability of an audio segment is calculated as the accumulation of frame level probabilities over all frames. Then, the audio event having the highest accumulated probability is determined as the segment level classification result. While in the voting method, classification is performed in frame level. The segment level event count is calculated as the number of frames classified with the corresponding event. The event with the maximum segment level event count is determined as the segment level classification result. Experimental results in **Fig. 5** and **Table 3** show that CNN3 model exhibits the highest performance, although the loss value of CNN1 is lower than that of CNN3. As the size of CNN1 model is relatively smaller than that of CNN3 it is concluded that CNN1 model has overfitted to the train dataset. It is shown that the probability accumulation method outperforms the voting method.

Table 3. Model test result

Model name	**Frame level accuracy (%)**	**Segment level accuracy (%)**	
		Prob. Acc.	**Voting**
DNN1	39.1	59.5	57.3
DNN2	37.3	55.3	53.1
DNN3	35.7	56.5	51.1
CNN1	50.3	78.0	74.7
CNN2	46.5	68.0	65.8
CNN3	57.2	81.5	78
CNN4	56.3	78.3	74.3

Confusion matrix of segment level test for CNN3 is shown in Fig. 6. Wind, car roads, and crowd are the classes that show relatively low performance. Six segments of wind are misclassified to sea waves and park, and six segments of car roads are misclassified to wind, and four segments of crowd are misclassified to children playing.



	air_conditioner	car_horn	children_playing	dog_bark	drilling	engine_idling	gun_shot	jackhammer	siren	street_music	baby_cry	boiling	rain	river	wind	bird	car_roads	cow	crowd	sea_wave	ship	horse	city_center	train	cafe/restaurant	grocery_store	park	metro_station	office	forest_path
air_conditioner	15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	2	0	1	0	0	0	0	0	0	0	0	0	0
car_horn	0	19	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
children_playing	0	0	14	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	2	1	0	0	0
dog_bark	0	0	0	17	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
drilling	2	0	0	0	16	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
engine_idling	1	0	0	0	0	17	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gun_shot	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jackhammer	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
siren	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
street_music	0	1	0	0	2	0	0	0	0	14	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
baby_cry	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
boiling	0	0	0	0	0	0	0	0	0	0	0	17	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
rain	0	0	0	0	0	0	2	0	0	0	0	0	16	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
river	1	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
wind	1	0	0	0	0	0	0	1	0	0	0	0	0	0	8	1	2	0	0	3	0	0	1	0	0	0	3	0	0	0
bird	0	0	1	0	1	0	0	0	0	0	0	0	0	0	3	14	0	0	0	0	0	0	0	0	0	0	1	0	0	0
car_roads	0	1	1	0	1	0	0	0	0	0	0	0	0	0	6	1	5	0	0	1	0	0	2	1	0	0	0	0	1	0
cow	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	1	0	0	0	1	0	0	1	0
crowd	0	1	4	0	0	0	0	0	2	1	0	0	2	0	0	0	1	0	5	0	0	0	1	0	1	0	0	0	2	0
sea_wave	0	0	0	0	0	0	1	0	0	0	0	0	1	0	3	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0
ship	0	0	0	0	0	0	0	2	0	0	0	1	1	0	1	0	1	0	0	0	13	1	0	0	0	0	0	0	0	0
horse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	16	0	1	0	0	0	1	1	0
city_center	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	1	0	0	0
train	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0
cafe/restaurant	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	5	0	0	0	0
grocery_store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0
park	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0
metro_station	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	19	0	0
office	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	16	0
forest_path	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20

Fig. 6. Confusion matrix of segment level test for CNN3 model

5. Conclusion

In this study, a CNN-based audio event classifier was proposed and its performance was verified through experimentation. Proposed system used the features of audio sound as an input image of CNN. Mel scale filter bank features were extracted from each frame, then the concatenated features over 40 consecutive frames were regarded as an input image. The event probabilities of output layer of CNN for all images in an audio segment were accumulated, then the audio event having the highest accumulated probability was determined to be the classification result. The performance of DNN-based and CNN-based classifiers with various model structures were measured. Experimental results exhibited a maximum performance of 81.5 %, which is approximately 20 % higher than the performance of the baseline classifier.

Acknowledgement

This work was supported by the ICT R&D program of MSIP/IITP. [2015-0-00225, Development of Media Application Framework based on Multi-modality which enables Personal Media Reconstruction]

References

[1] K. Kim and H. Kim, "Storytelling Strategy of Visual-Image Contents base on Rhetoric Metaphors," *Journal of Digital Content Society*, vol. 14, no. 4, pp. 481-491, December, 2013. Article (CrossRef Link)

[2] L. Lu, H. Jiang and H. Zhang, "A robust audio classification and segmentation method," in *Proc. of ACM International Conference on Multimedia*, pp. 203-211, September 30-October 5, 2001. Article (CrossRef Link)

[3] M. Xu, N. Maddage, C. Xu, M. Kankanhalli and Q. Tian, "Creating audio keywords for event detection in soccer video," in *Proc. of IEEE International Conference on Multimedia and Expo*, pp.281-284, July 6-9, 2003. Article (CrossRef Link)

[4] W. Cheng, W. Chu and J. Wu, "Semantic context detection based on hierarchical audio models," in *Proc. of ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp.109-115, November 7-7, 2003. Article (CrossRef Link)

[5] H. Lee, P. Pham, Y. Largman and Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. of Advances in Neural Information Processing Systems*, pp.1096-1104, December 7-10, 2009.

[6] Y. Bengio and Y. LeCun, "Large-scale Kernel Machines," *MIT Press*, 2007.

[7] J. Portelo, M. Bugalho, I. Trancoso, J. Neto, A. Abad and A. Serralheiro, "Non-speech audio event detection," in *Proc. of Internationa Conference on Acoustics, Speech and Signal Processing*, pp.1973-1976, April 19-24, 2009. Article (CrossRef Link)

[8] L. Ballan, A. Bazzica and M. Bertini, A. Bimbo, and G. Serra, "Deep networks for audio event classification in soccer videos," in *Proc. of International Conference on Multimedia and Expo*, pp.474-477, June 28-3, 2009. Article (CrossRef Link)

[9] T. Heittola, A. Mesaros, A. Eronen and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol.1, pp.1-13, January, 2013. Article (CrossRef Link)

[10] K. Zvi and T. Orith, "Audio event classification using deep neural networks," in *Proc. of Interspeech*, pp.1482-1486, August 25-29, 2013.

[11] S. Downie, et al., "The Music Information Retrieval Evaluation eXchange: Some observations and insights," *Advances in Music Information Retrieval*, pp. 93-115, 2010. Article (CrossRef Link)

[12] R. Malkin, "Multimodal Technologies for Perception of Humans," *Springer*, pp. 323-330, 2007. Article (CrossRef Link)

[13] F. Smeaton, et al., "Evaluation campaigns and TRECVid," in *Proc. of ACM International Workshop on Multimedia Information Retrieval*, pp. 321-330, 2006. Article (CrossRef Link)

[14] E. Vincent, et al., "The signal separation evaluation campaign (2007–2010): Achievements and remaining challenges," *Signal Processing*, vol. 82, no. 8, pp. 1928-1936, 2012. Article (CrossRef Link)

[15] H. Larochelle, et al., "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proc. of International Conference on Machine Learning*, pp.473-480, 2007. Article (CrossRef Link)

[16] M. Lim and J. Kim, "Audio Event Classification Using Deep Neural Networks," *Phonetics and Speech Sciences*, vol. 7, no. 4, pp.27-33, January, 2015. Article (CrossRef Link)

[17] J. Salamon, C. Jacoby and J. Bello, "A dataset and taxonomy for urban sound research," in *Proc. of ACM International Conference on Multimedia*, pp.1041-1044, November 3-7, 2014. Article (CrossRef Link)

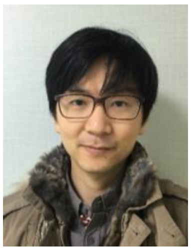
[18] M. Slaney, "Semantic-audio retrieval," in *Proc. of International Conference on Acoustics, Speech and Signal Processing*, pp.1408-1411, May 13-17, 2002. Article (CrossRef Link)

[19] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. of 24th European Signal Processing Conference*, pp. 1128-1132, 2016. Article (CrossRef Link)

[20] S. Young, G. Evermann, M. Gales and P. Woodland, "The HTK book (for HTK version 3.4)," *Entropic Cambridge Research Laboratory*, 2006.

[21] M. Abadi, A. Agarwal, et al, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016, Preprint at Article (CrossRef Link).

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp.436-444, May, 2015. Article (CrossRef Link)



Minkyu Lim received his B.E. degree in Computer Science and Engineering as well as in Mechanical Engineering from Sogang University in 2008. He also received his M.E. degree in Computer Science and Engineering from Sogang University in 2010. He is currently pursuing a Ph.D. degree in Computer Science and Engineering at Sogang University. His research interests include speech recognition and spoken multimedia content search.

Donghyun Lee received his B.E. degree in Computer Science and Engineering from Sogang University in 2013. He is currently pursuing a Ph.D. degree in Computer Science and Engineering at Sogang University. His research interests include speech recognition and spoken multimedia content search.

Hosung Park received his B.E. degree in Computer Science and Engineering from Handong Global University in 2016. He is currently pursuing a M.E. degree in Computer Science and Engineering at Sogang University. His research interests include speech recognition and spoken multimedia content search.

Yoseb Kang received his B.E. degree in Mathematics as well as in Economics from Sogang University in 2017. He is currently pursuing a M.E. degree in Computer Science and Engineering at Sogang University. His research interests include speech recognition and spoken multimedia content search.

Junseok Oh received his B.E. degree in Computer Science and Engineering from Sogang University in 2017. He is currently pursuing a M.E. degree in Computer Science and Engineering at Sogang University. His research interests include speech recognition and spoken multimedia content search.

Jeong-sik Park received his B.E. degree in Computer Science from Ajou University, South Korea in 2001 and his M.E. and Ph.D. degree in Computer Science from KAIST (Korea Advanced Institute of Science and Technology) in 2003 and 2010, respectively. From 2010 to 2011, he was a Post-Doc. researcher in the Computer Science Department, KAIST. He had been a faculty member in Mokwon University and Yeungnam University. He is now an assistant professor in the Department of English Linguistics & Language Technology, Hankuk University of Foreign Studies. His research interests include speech emotion recognition, speech recognition, speech enhancement, and voice interface for human-computer interaction.

Gil-Jin Jang is an assistant professor at Kyungpook National University, South Korea. He received his B.S., M.S., and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, South Korea in 1997, 1999, and 2004, respectively. From 2004 to 2006 he was a research staff at Samsung Advanced Institute of Technology and from 2006 to 2007 he worked as a research engineer at Softmax, Inc. in San Diego. From 2008 to 2009, he joined Hamilton Glaucoma center at University of California, San Diego as a postdoctoral employee. He had been a faculty member in UNIST. His research interests include acoustic signal processing, pattern recognition, speech recognition and enhancement, and biomedical signal engineering.

Ji-Hwan Kim received the B.E. and M.E. degrees in Computer Science from KAIST (Korea Advanced Institute of Science and Technology) in 1996 and 1998 respectively and Ph.D. degree in Engineering from the University of Cambridge in 2001. From 2001 to 2007, he was a chief research engineer and a senior research engineer in LG Electronics Institute of Technology, where he was engaged in development of speech recognizers for mobile devices. In 2004, he was a visiting scientist in MIT Media Lab. Since 2007, he has been a faculty member in the Department of Computer Science and Engineering, Sogang University. Currently, he is a full professor. His research interests include spoken multimedia content search, speech recognition for embedded systems and dialogue understanding.