DNN-BASED AUDIO SCENE CLASSIFICATION FOR DCASE 2017: DUAL INPUT FEATURES, BALANCING COST, AND STOCHASTIC DATA DUPLICATION

Jee-Weon Jung, Hee-Soo Heo, IL-Ho Yang, Sung-Hyun Yoon, Hye-Jin Shim, and Ha-Jin Yu[†]

School of Computer Science, University of Seoul, South Korea

ABSTRACT

In this study, we explored DNN-based audio scene classification systems with dual input features. Dual input features take advantage of simultaneously utilizing two features with different levels of abstraction as inputs: a frame-level mel-filterbank feature and segment-level identity vector. A new fine-tune cost that solves the drawback of dual input features was developed, as well as a data duplication method that enables DNN to clearly discriminate frequently misclassified classes. Combining the proposed methods with the latest DNN techniques such as residual learning achieved a fold-wise accuracy of 95.9% for the validation set and 70.6% for the evaluation set provided by the Detection and Classification of Acoustic Scenes and Events community.

Index Terms— audio scene classification, DNN, dual input feature, balancing cost, data duplication, residual learning

1. INTRODUCTION

Vast amounts of information can be acquired from an audio segment, including the surrounding environment. Audio scene classification is a task of classifying the surrounding environment based on a given audio segment. Various approaches have been explored for successful audio scene classification. For example, non-negative matrix factorization (NMF) [1], Gaussian mixture model (GMM) [2] were used in the 2016 Detection and Classification of Acoustic Scenes and Events (DCASE) competition.

In this study, we concentrated on exploiting deep learning-based systems. Recent advances in deep learning have made deep neural networks (DNNs) a state-of-the-art system for many tasks [3], [4]. We exploited approaches used in other tasks, such as image recognition and speaker verification. For example, residual network architecture [5] is a state-of-the-art system for image recognition. The identity vector (i-vector) [6], which composes state-of-the-art systems in speaker verification, is extracted from an audio segment and used as one of the two input features for the DNN classifier. The widely used mel-filterbank feature is also used simultaneously as the other input feature.

The remainder of this paper is organized as follows. Section 2 describes the proposed techniques. Subsection 2.1 describes the baseline, and the other subsections describe the methods that are stacked to compose the systems submitted to the 2017 DCASE challenge. Section 3 describes the experimental settings and configurations, along with the results and analysis.

The first and second author in this paper have the same contribution.

[†] Corresponding author

2. SYSTEM DESCRIPTION

The methods presented in this section were separately or simultaneously applied to our DNN baseline to compose four submitted systems. System 1 with dual input features was our baseline. The methods in each subsection were stacked to compose Systems 2 to 4. The newly defined balancing cost was applied to System 1 to create System 2. Stochastic data duplication based on the classification accuracy of the development set was added to System 2 to create System 3. The DNN architecture of System 3 was changed from multi-layer perceptron (MLP) to a residual network to create System 4 [5].

2.1. Dual Input Features (System 1)

Two different features were utilized as inputs for the DNN: the mel-filterbank feature (frame-level) and the i-vector (segment-level). Description about the mel-filterbank feature is omitted because it is a widely used method.

The i-vector (identity vector) [6], is an segment-level feature and represents the core identity of an audio segment. Thus, a single vector is extracted from each audio segment regardless of its length. One i-vector and several mel-filterbank features extracted from one audio segment were connected and used as the DNN input. In this case, one i-vector is duplicated and connected to the mel-filterbank features extracted in frame unit. Figure 1 shows an overview of the dual feature-based system. Although i-vectors were originally designed to represent the identity of a speaker, recent research has shown that they can be used for tasks related to scene classification [7]. Thus, we utilized i-vectors for audio scene classification.

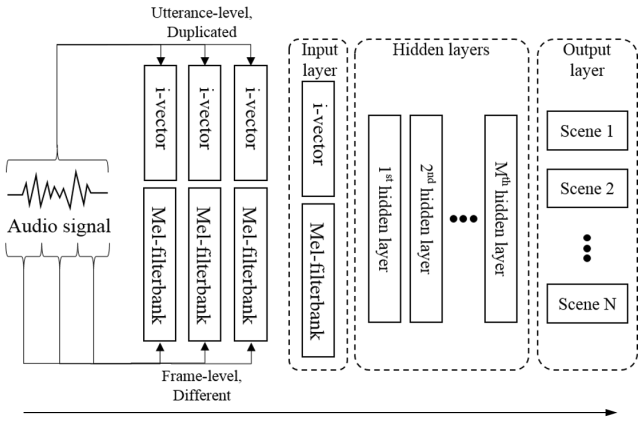

Figure 1: Illustration of mel-filterbank features and i-vectors being simultaneously used as input to the DNN.

In this study, the mel-filterbank featurse and i-vectors were both used as inputs for the DNN classifier with the expectation of synergy between two features with different levels of abstraction: frame and segment, respectively. However, in our early experiments with the DCASE 2016 challenge dataset, the classification accuracy of the system using dual input features (84.1%) was almost the same as that of a system trained only with i-vectors (83.8%).

A performance gain could be achieved by pre-training the DNN classifier with only the mel-filterbank features and then fine-tuning it with both input features. The mel-filterbank feature was pre-trained by zero-masking the i-vector features and fine-tuning the network a certain number of times while maintaining the DNN architecture. It was possible to improve the scene classification accuracy of the DNN by applying the pre-training technique to System 1. However, applying the pre-training technique requires a large amount of training time and knowledge of the characteristics of input features.

2.2. Balancing Cost (System 2)

We discovered an unexpected characteristic of the DNN by visualizing the weight matrix between the input layer and the first hidden layer. The DNN tended to utilize only the i-vector and neglect the mel-filterbank features when the two features were input simultaneously. The absolute values for the weights of the network connecting the mel-filterbank features to the first hidden layer were almost zero. In contrast, the absolute value for the weight of the network connecting the i-vector to the first hidden layer was relatively high. We hypothesized that this phenomenon occurred because the i-vector, which is an segment-level feature, was easier to utilize. In contrast, the mel-filterbank feature is a frame-level feature and was harder to utilize.

To solve the problem of the network neglecting the mel-filterbank features, we added BF_1 and BF_2 to the network's negative log likelihood (NLL) fine-tune cost in (1) with α and β as scale factors. Equation (2) corresponds to the average effectiveness of one node from the input feature to all nodes in the first hidden layer. Here, $W_{x,y}$ means the weight connections between x^{th} node in the input layer and y^{th} node in the first hidden layer where X, Y refer to the number of nodes in the input layer and the first hidden layer, respectively. Thus, BF_1 in (3) represents the variance of influence for each element in the input layer. A lower BF_1 means that the input feature's elements are evenly utilized. By adding BF_1 to the objective function, each element of the input feature is forced to have similar effectiveness on the next layer.

$$cost = NLL + \alpha * BF_1(W) + \beta * BF_2(W) \tag{1}$$

$$f_1(W_x) = \frac{1}{Y}\sum_{y=1}^{Y}|W_{x,y}| \tag{2}$$

$$BF_1(W) = Var(f_1(W_1), f_1(W_2), ..., f_1(W_X)) \tag{3}$$

However, the easiest way to make BF_1 equal to zero is by converging all weights to zero. To prevent this, BF_2 in (5) was introduced. For every mini-batch, the average scale of the weight matrix , W^{cur}, is compared with the scale of initial weight matrix, W^{init}. By applying ReLU function, BF_2 is active only if the scale of weight matrix has decreased. Thus, BF_2 prevents the weights from converging to zero.

$$f_2(W) = \frac{1}{X}\frac{1}{Y}\sum_{x=1}^{X}\sum_{y=1}^{Y}W_{x,y} \tag{4}$$

$$BF_2(W) = ReLU(f_2(W^{init}) - f_2(W^{cur})) \tag{5}$$

With the added terms to balance mel-filterbank features and i-vector, the two input features were evenly used judging by the absolute value of the weight matrix.

2.3. Stochastic data duplication (System 3)

A confusion matrix was used to analyze the performance of the subsystems. A combination of three channels that we divided and four cross-validation folds provided by the DCASE community generates 12 subsystems. Details regarding the channels are addressed in subsection 2.5. Figure 2 shows the confusion matrix generated by 12 subsystems from the classification experiment. Diagonal elements for the confusion matrix are omitted in Figure 2 for better visualization of misclassified audio segments. The results confirmed that the misclassification of each subsystem was concentrated in few classes unique to each subsystem. Thus, we devised a simple method to emphasize specific audio scenes in the training phase.

The suggested method (i.e., stochastic data duplication) duplicates each scene's train dataset proportional to the number of misclassified audio segments. It was applied after every epoch during the training phase based on class-wise accuracy with the validation set. Equations (5) and (6) describes how stochastic data duplication is conducted. C corresponds to the confusion matrix. $C_{j,k}$ is the number of misclassified audio segments where scene k was classified as scene j. E_k refers to the number of misclassified audio segments for scene k. In (6), A_k, between 0 and 1, is the proportion of data from scene k that is duplicated where K refers to the set of audio scenes.

$$E_k = \sum_j C_{j,k} - C_{k,k} \tag{6}$$

$$A_k = \frac{E_k}{\sum_i^K E_i} \tag{7}$$

Strictly speaking in terms of validation accuracy, this approach may not be appropriate because validation result itself is exploited. Thus, the classification accuracy derived from the validation set was described only for reference. However, from the perspective of the actual DCASE 2017 challenge, the validation set is part of the development set. Therefore, the evaluation result with stochastic data duplication applied is valid.

2.4. Residual learning (System 4)

The number of hidden layers has always been a core hyperparameter in deep learning with a major effect on system performance. A residual network was proposed in [5] to resolve this problem. As illustrated on the right side of Figure 3, the residual network is composed of several residual blocks. Each residual block has an identity mapping connection. In identity mapping [8], an input x is directly mapped to the output, and $F(x, W)$ calculate the residual only, where W is the weight matrix. Based on the identity mapping connection, the input can easily be identically mapped to the output by making the weight matrix into zeros. Thus, as far as the hardware supports it, higher performance is expected with deeper networks because unneeded residual blocks are trained for identity mappings.

$$y = F(x, W) + x \tag{8}$$

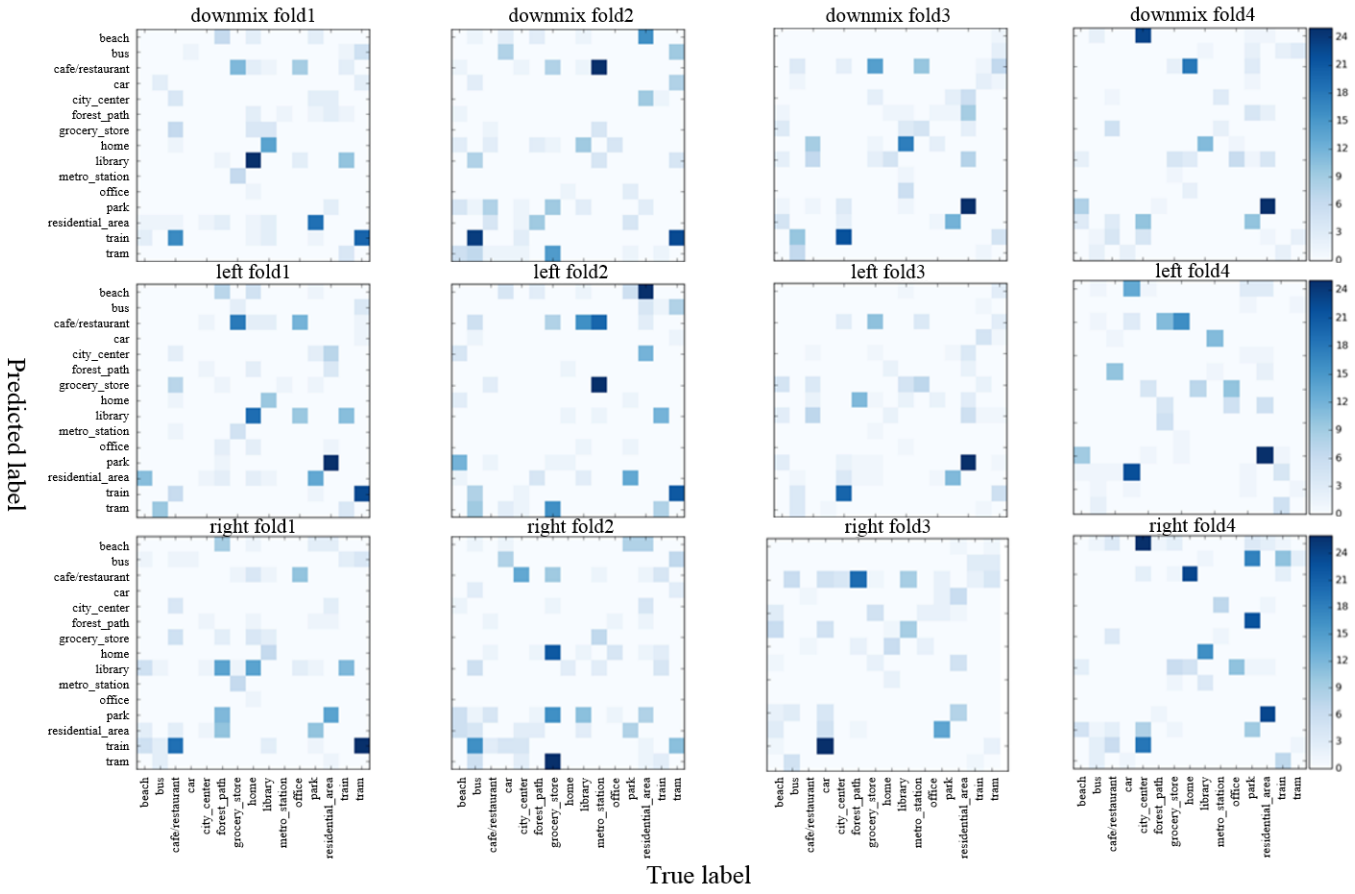

Figure 2: Confusion matrices for all channels and folds.

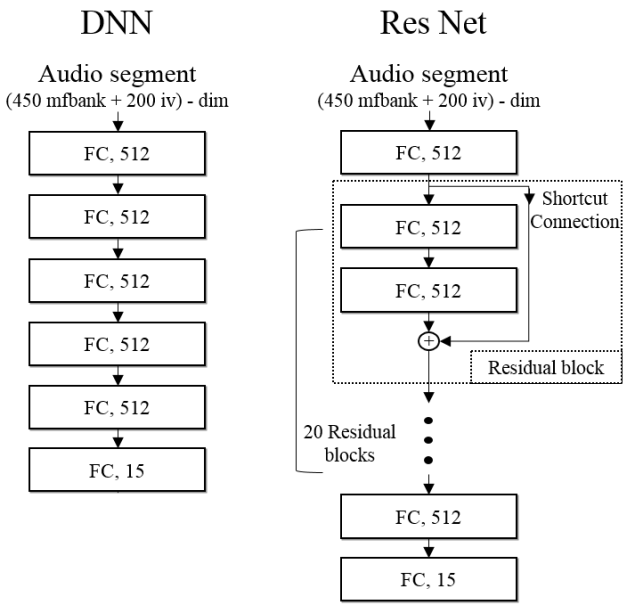

Figure 3: Network architectures. **Left**: DNN of 5 fully-connected hidden layers. **Right**: Residual network of 42 hidden layers (20 blocks + 2 fully-connected).

In [8], a different residual block composition was proposed where the sequence of weight operation, batch normalization, and then the activation function is replaced by a sequential implementation of batch normalization, activation function and then weight summation. The new form, which is called full pre-activation, can preserve clean information paths and gives a performance gain. This technique is adopted for System 4 by changing the network of System 3 to a 42-layer MLP.

2.5. Score-level ensemble

Audio segments from the DCASE 2017 challenge were provided at a 44.1kHz sampling rate and 24-bit resolution, in stereo. In many other studies, stereo audio is normally converted to mono audio by averaging two channels at the sample-level before processing. However based on former studies for the DCASE 2016 challenge [9], all systems in this study, including System 1 (baseline), extracted three different channels from one audio segment: the left and right channels of the stereo audio segment and the converted mono. Then, feature extraction and DNN classifier training were conducted sequentially for each channel. During the verification and evaluation stage, we use the ensemble of the decision scores of the three independent systems to make the final decision for an audio segment.

The results showed that a performance gain was achieved when ensemble was conducted on networks of different channels. Table 1

shows the classification accuracy of individual channels and when ensemble of the three channels is used.

Table 1: Classification accuracy(%) of individual channels and score-level ensemble network using the dual input feature model (System 1).

Channel	Fold 1	Fold 2	Fold 3	Fold 4	Average
converted	84.8	81.8	80.9	83.7	82.8
left	82.2	81.9	81.4	82.7	82.0
right	83.5	81.6	82.3	79.8	81.8
Ensemble	86.2	86.1	84.3	85.2	85.5

3. EXPERIMENTS AND RESULTS

The systems were implemented with Theano [10], [11], which is a Python library for deep learning. The open-source toolkit Kaldi [12] was used to extract i-vectors.

3.1. Feature Configuration

40-dimensional mel-filterbank feature was extracted by using 25ms windows with 10ms shift, following [13]. Linear discriminant analysis (LDA) [14] was used to reduce the dimension of mel-filterbank features to 10. The total feature of a frame is the concatenation of the mel-filterbank feature vectors of the frame and 22 frames before and after the current frame and an i-vector of the segment.

A diagonal GMM with 1024 components is trained with 60-dimensional mel-frequency cepstral coefficients, and a total variability matrix that can extract an i-vector of 200 dimensions was trained for 10 iterations. A 450-dimensional mel-filterbank features ($10 \times (22 + 1 + 22)$) and a 200-dimensional i-vector were concatenated to form 650-dimensional input feature for the DNN classifier.

3.2. Network Configuration

For Systems 1 to 3, MLP has four hidden layers, each having 512 nodes. For System 4, the MLP has 42 hidden layers, each having 512 nodes with a residual connection for every two layers.

The L-2 weight decay [15] with a lambda of 10^{-4} is applied. Dropout [16] was applied to all systems with 20% dropout at the first hidden layer and 50% for the rest of the hidden layers except for System 4. The exclusion of dropout for System 4 follows the practices in [5], [8]. Batch normalization [17] and learning rate decay following implementation in [18] were also utilized. α and β in our balancing cost were set to 1000 and 100 respectively.

For a residual network, the first and last hidden layers are normal fully-connected layers without a residual connection. In addition, when full pre-activation is applied, the batch normalization and activation functions should be excluded for the first layer in the first residual block. This is to avoid duplicate application of the batch normalization and activation function.

3.3. Results and Analysis

Table 2 presents the validation results for the four-fold cross validation of our submitted systems.

In System 2, the balancing cost reduced the classification accuracy by 0.5%p compared with System 1. However, the balancing cost does not require pre-training with only mel-filterbank features. Thus, it can reduce the complexity of the system.

Applying stochastic data duplication led to a recognizable performance gain for all folds except fold 1. However, because the results of the validation set is utilized, strictly speaking in the perspective of validation set, this may not be considered a fair experiment. The performances of Systems 3 and 4 with validation set are therefore presented for reference only. The actual performance gain from stochastic data duplication in the evaluation is addressed in Table 2.

Table 2: Classification accuracy(%) of 4-fold average on validation set and evaluation set.

System #	Validation set	Evaluation set
System 1	85.5	67.0
System 2	85.1	66.2
System 3	95.5	67.3
System 4	**95.9**	**70.6**

- System 1: dual input feature
- System 2: dual input feature + balancing cost
- System 3: dual input feature + balancing cost + stochastic data duplication
- System 4: dual input feature + balancing cost + stochastic data duplication + residual network

4. CONCLUSION

In this paper, the latest DNN-based approaches was evaluated by application to audio scene classification with proposed approaches applied. System 1(baseline), which used dual input features, showed 67.0% classification accuracy with the evaluation dataset, 6.0%p higher than the classification accuracy of the DCASE baseline system. Necessity of pre-training was resolved with balancing cost in System 2. A technique for further training the DNN for frequently misclassified classes was applied to System 3. System 4 which applied proposed approaches in Systems 1 through 3 and residual network achieved a classification accuracy of 70.6%, showing that the proposed approaches are valid.

5. ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(2017R1A2B4011609)

6. REFERENCES

[1] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

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

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[6] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[7] B. Elizalde, H. Lei, G. Friedland, and N. Peters, "An i-vector based approach for audio scene detection," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[9] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "Cp-jku submissions for dcase-2016: A hybrid approach using binaural i-vectors and deep convolutional neural networks," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[10] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: A cpu and gpu math compiler in python," in *Proc. 9th Python in Science Conf*, 2010, pp. 1–7.

[11] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: new features and speed improvements," *arXiv preprint arXiv:1211.5590*, 2012.

[12] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[13] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[14] J. Qin and A. Waibel, "Application of lda to speaker recognition," in *Interspeech*, 2000.

[15] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems(NIPS)*, 1992.

[16] N. Srivastava, G. 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, pp. 1929–1958, 2014.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[18] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," *arXiv preprint arXiv:1703.01789*, 2017.