FUSION MODEL BASED ON CONVOLUTIONAL NEURAL NETWORKS WITH TWO FEATURES FOR ACOUSTIC SCENE CLASSIFICATION

Jinwei Xu, Yang Zhao, Jingfei Jiang, Yong Dou, Zhiqiang Liu, Kai Chen

Science and Technology on Parallel and Distributed Processing Laboratory,
National University of Defense Technology, Changsha 410000, China

ABSTRACT

This report describes two submissions for Task 1 (audio scene classification) of DCASE-2017 challenge of PDL team. We propose two different approaches for Task 1. First, we propose a new convolutional neural network (CNN) architecture trained on frame-level features such as mel-frequency cepstral coefficient (MFCC) of audio data. Second, we propose a late fusion of the proposed CNN trained with two different features, namely, MFCCs and spectrograms. We report the performance of our proposed methods on the cross-validation setup for Task 1 of DCASE-2017 challenge.

Index Terms— audio scene classification, convolutional neural network, mel-frequency cepstral coefficient, spectrograms image features, inception-resnet-v1, late fusion

1. INTRODUCTION

When audio scene is discussed, a place with mixed sound is always referred to. The sound of the audio scene is mixed with many sounds, such as chirm, bicker, car siren, and others. If we assume that the audio scene was generated by a special speaker, we can apply a speaker identification algorithm to solve an audio scene classification task. Many speaker identification algorithms can be used in audio scene classification tasks. Thus, in the previous challenges, the participants proposed Gaussian mixture models, i-vector models, and deep neural network (DNN) models to solve classification tasks.

With the development of deep learning algorithms, an increasing number of DNNs are applied to DCASE challenges. Last year, Valenti proposed a CNN architecture with 86.2% accuracy [1]. While Eghbal-Zadeh proposed a VGG-style convolutional neural network with 83.3% accuracy [2]. Inspired by this, we propose a novel convolutional neural network referred to Inception-ResNet-v1, which is proposed by Szegedy in AAAI-17[3]. Inception-ResNet-v1 has exhibited a perfect performance on the non-blacklisted subset of the validation set of ILSVRC 2012. Moreover, the gained Top-1 Error is approximately 18.8%, which is significantly less than that of VGG.

In this report, two methods are described for Task 1 of DCASE-2017 challenge. We provide the performance of our methods on the cross-validation setup for Task 1 of DCASE-2017 [4]. First, we propose a novel CNN model referred to Inception-ResNet-v1. The input of the CNN model is MFCCs of the audio dataset. Then, we find that, the same CNN model with different input features can have various effects on the same data. Thus, we fusion two scores of the models with two different input features, which constitute our second submission.

The rest of this report is organized as follows. In Section 2, the Inception-ResNet-v1 is described. The proposed methods and experiments setting are explained in Section 3. In Section 4, the results of audio scene classification on the provided dataset and cross-validation splits are presented. Finally, we conclude this report in Section 5.

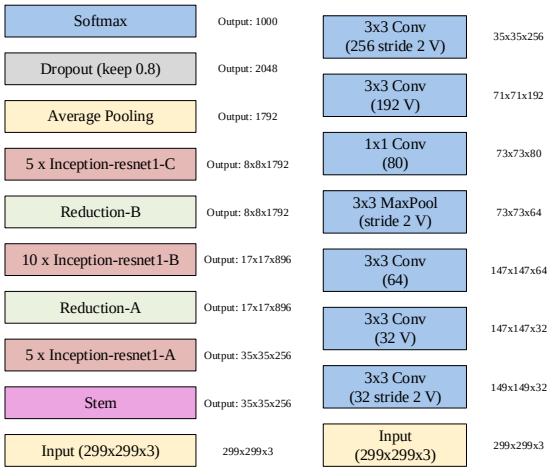

Figure 1: On the left is the overall schema for Inception-ResNet-v1 network. And on the right is the stem of Inception-ResNet-v1. V denotes the use of 'Valid' padding.

2. INCEPTION-RESNET-V1

Inception-ResNet-v1 was proposed by Szegedy in AAAI-17, and Szegedy et al. study the combination of two of the most recent ideas: Residual connections [5] and the latest revised version of Inception architecture [6]. [5] proposed that the residual connection architectures are inherently important for very deep convolutional neural networks. And to get very deep inception network, it is nature to replace the filter concatenation stage of inception architecture with residual connection. Thus, Szegedy proposed Inception-ResNet-v1 network.

The full configuration of the Inception-Resnet-v1 network is outlined in Figure 1 which contains the overall schema and stem configuration. Moreover, Figure 2 that has the detailed configuration of the interior modules.

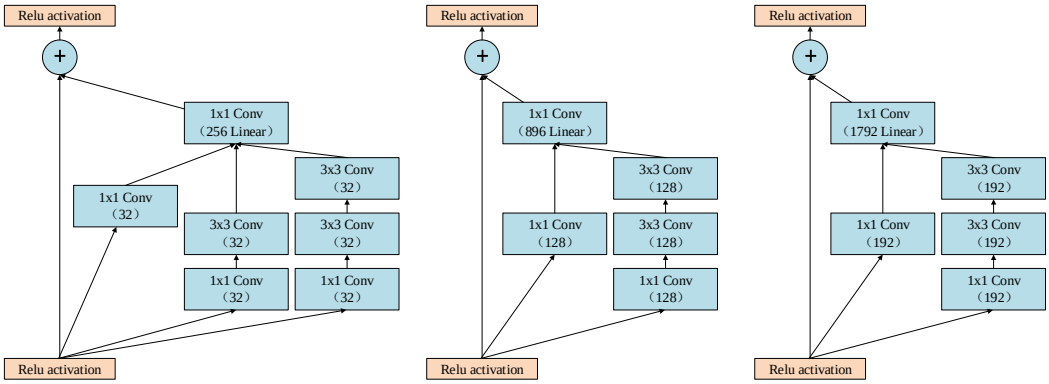

Figure 2: The schema for interior grid modules of the Inception-ResNet-v1 network. The 35 × 35, 17 × 17 and 8 × 8 grid modules are depicted from left to right. There are the Inception-A, Inception-B, and Inception-C blocks of the schema on the left Figure 1.

3. PROPOSED METHOD

In this section we describe our methods and the architecture chosen for the proposed system. The block diagram of the proposed system is illustrated in Figure 3.

3.1. framework

Our system consists of two CNN model. The input feature of the upper CNN model is MFCCs and the input of the below one is SIFs. We train the two CNN model separately and fuse the scores after the softmax layer. To fuse those scores, we suggest to compute the mean of them.

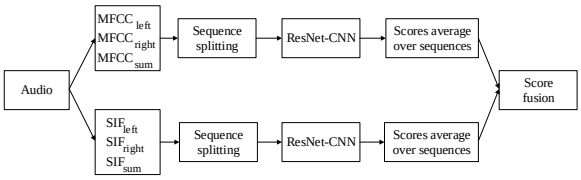

Figure 3: Block diagram of the proposed system.

3.2. Feature representation and preprocessing

We use two types of feature representation, MFCCs and SIFs, in our proposed system. First, to calculate the SIF feature, we apply short-time Fourier transform (STFT) over windows of 40 ms of audio with 50% overlap and Hamming windowing. After conducting STFT, we downsample the feature from 2048 to 99 bins in the frequency domain. Thereafter, we obtain the SIF feature representation. Second, to obtain the MFCCs, we square the absolute value of each bin and apply a 32-band mel-scale filter bank in the range of 0C44 kHz. Then, the logarithmic conversion of the mel energies are computed. Finally, we use DCT to obtain 33 cepstral coefficients. However, these MFCCs can only reflect the static information. We calculate deltas and double deltas of the MFCCs to obtain the dynamic information. Third, we split these features into shorter ones to adapt to the input size of the CNN model and gain better accuracy. At the end of preprocessing, the input to the CNN is a 99×99 (2 seconds long) matrix that can be displayed as an image.

To enrich the audio material, we extracted the features from left, right, and average channels, as shown in Figure 4. Then, we can acquire a 99 × 99 × 3 matrix similar to a 3-channel image with RGB channels. All preprocessing has been implemented with the Matlab toolbox Voicebox [10].

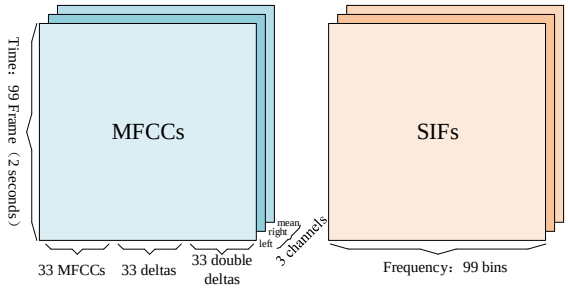

Figure 4: The input of proposed CNN model. On the left is Mel-Frequency Cepstral Coefficients (MFCCs), while spectrograms image features (SIFs) is shown on the right. Both of them include three channel, left, right and mean channel.

3.3. Proposed CNN Model

This section describes the proposed CNN model. Our CNN model is designed and referred to as Inception-resnet-v1, which is processed on an ImageNet data set. The data set typically consists of 1,000 classes and millions of pictures. However, our data set consists of only 15 classes and approximately 5,000 pieces of audio material. If we use Inception-resnet-v1 without modification, then the model becomes redundant for our dataset and causes overfitting. Thus, the Inception-resnet-v1 model was simplified. Figure 5 illustrates our proposed CNN model. The input size of our model is 99 × 99 × 3, and then we tailored the stem from six convolutional layers to four. Third, we removed Inception-resnet-A and

Table 1: Hyper-parameter of the proposed CNN networks.

Hyper-parameter	Value
Initial learning rate	0.01
L2-weight decay penalty	0.0001
Center loss penalty	0.001
Dropout rate	0.8
Batch normalization parameter	0.00004
Mini-batch size	50

Reduction-A, and added a fully-connected layer to reduce the output node to 64. Finally, we modified the output node of the softmax layer to 15. Our design simplified the network parameters to improve accuracy.

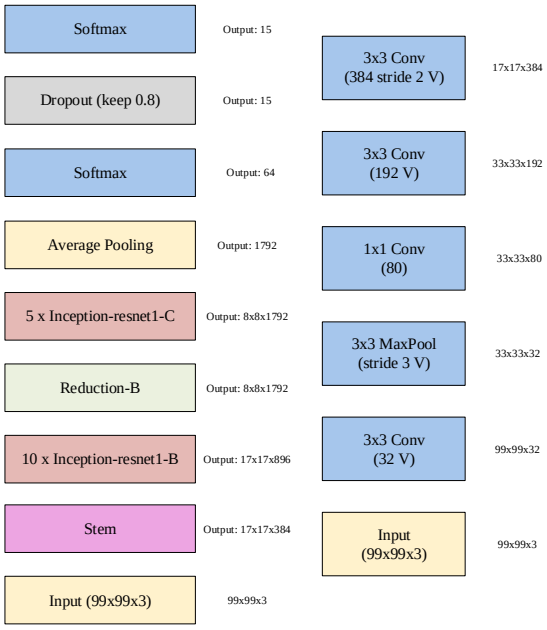

Figure 5: The proposed convolutional neural network. On the left is the overall schema for the proposed network. And on the right is the stem.

The proposed CNN architecture involves various hyper-parameters that are specified in Table 1. The parameters of our models were optimized with mini-batch stochastic gradient descent. The networks were trained using a training set and evaluated using an evaluation set in every fold with a mini-batch size of 50. We started training with an initial learning rate of 0.01 and for every 8 epochs, it decays with a rate of 0.95. We applied a L2-weight decay penalty of 0.0001 and dropout of 0.8 to avoid overfitting. To reduce training time, we applied batch normalization and the parameter was 0.00004. To improve the accuracy, we added center loss to the loss functions and the penalty of the center loss was 0.001. Finally, the model was trained on a cross-validation data set and we trained 4 models for 4 folds with one type of feature. Finally, we acquired 8 models to test the evaluation data.

Table 2: Comparing the performance of our CNN model and i-vector model.

	fold1 acc	fold2 acc	fold3 acc	fold4 acc	average acc
MFCC-2s	82.87	79.90	80.55	86.10	82.36
SIF-2s	82.58	79.32	74.46	79.11	78.87
MFCC-10s	85.38	82.05	**83.68**	**90.09**	**85.30**
SIF-10s	**86.24**	**82.56**	78.80	83.33	82.73
i-vector	81.03	79.03	78.01	82.99	80.27

4. EXPERIMENT

This section describes the experiment setup, experiment results, and comparisons with other models. An analysis of our proposed system is also presented in this section. Our experiment platform is GTX1080Ti and all the results are implemented based on TensorFlow.

First, we performed some confirmatory trail based on cross-validation dataset. Table 2 illustrates the classification accuracy of two types of CNN models based on MFCC and SIF features. The table also presents a comparison with the i-vector model, which is implemented and referred to the model in Eghbal. The UBMs with 256 Gaussian components on MFCC features were trained and the UBM, T matrix, LDA, and WCCN projections are trained on the training portion of each cross-validation split. The dimension of the i-vectors is set to 400 and cosine scoring is applied to the i-vector model. The MFCC features include 3 parts, 23 MFCCs (without 0th MFCC) with 20-ms frame length and without overlap, 18 MFCC deltas and 20 MFCC double deltas (both including the 0th) with 60-ms frame length and 40-ms overlap. To obtain a good result, 32 triangle-shaped mel-scaled filters in the range of 0C44 kHz were used. Notably, the length of the audio data of the CNN models is 2s. As Table 2 shows, the CNN model with MFCC features as its input has a higher classification accuracy than the i-vector model. However, the CNN model with SIF feature as its input has a comparable, or even worse, result relative to the i-vector model. Nevertheless, the input data of the i-vector model is a complete sound bite (10 seconds) that contains all information. Thus, we can obtain the mean of our scores from the same sound bite to gain good classification accuracy. MFCC-10s and SIF-10s show the result after obtaining the mean. As shown, MFCC-10s and SIF-10s have higher accuracy than i-vector when the input data are the same.

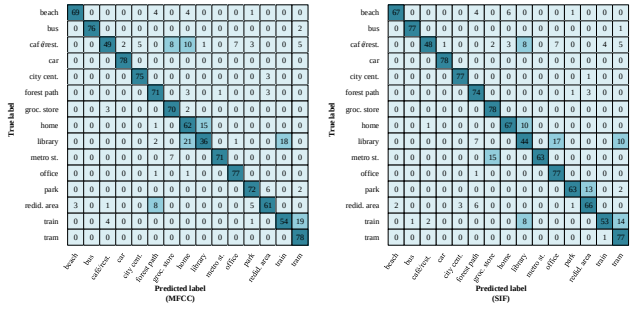

Figure 6: Confusion matrix for the proposed CNN evaluated on fold1.

Figures 5C8 show a confusion matrix for the proposed CNN

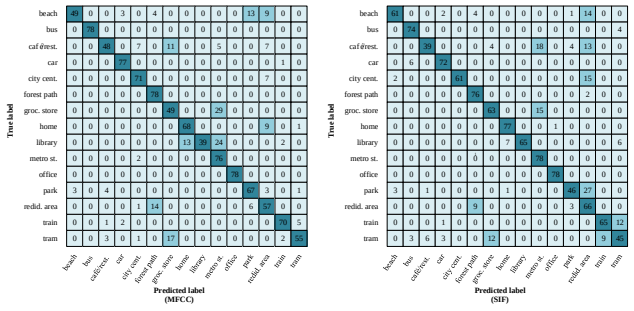

Figure 7: Confusion matrix for the proposed CNN evaluated on fold2.

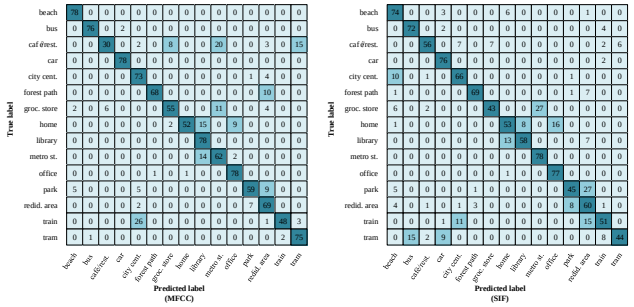

Figure 8: Confusion matrix for the proposed CNN evaluated on fold3.

evaluated on the four folds. As shown, the classification result of the two models is different for the same class. The CNN model based on the MFCC feature can obtain a good result for some classes and the one based on the SIF feature obtained a good result for several others. Therefore, we combine the two models to ensure higher classification accuracy.

Table 3 illustrates the accuracy of the fusion model. As shown, the fusion can acquire a better result than those of the two separate models. Nevertheless, the effect of fusion is extremely different for various data sets. For example, the fusion for fold2 and fold3 can gain an obvious effect with approximately 3% increase, but for fold1 and fold4, the effect is not obvious. We can obtain the reason by comparing their confusion matrix. For fold2 and fold3, the classification result of the two CNN models is extremely different. Conversely, for fold1 and fold4, the classification performance is particularly a good model that can conceal most of the erroneous results of the other model, but its erroneous result cannot be avoided by the other one. Therefore, the fusion cannot improve the classification accuracy.

Table 3: Table 5: Audio scene classification accuracy on the provided DCASE-2017 test set with provided cross-validation splits.

	fold1 acc	**fold2 acc**	**fold3 acc**	**fold4 acc**	**average acc**
MFCC	85.38	82.05	83.68	90.09	85.30
SIF	86.24	82.56	78.80	83.33	82.73
MFCC+SIF	**86.75**	**86.07**	**85.81**	**90.94**	**87.39**

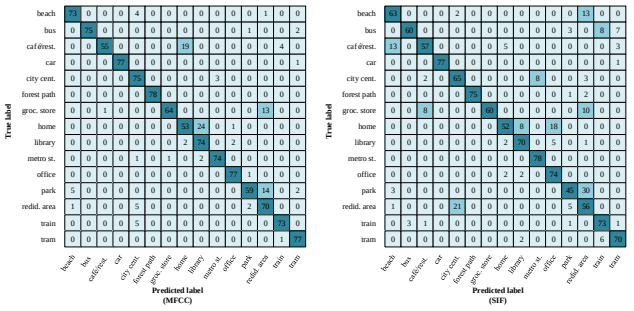

Figure 9: Confusion matrix for the proposed CNN evaluated on fold4.

Finally, based on the experiments, we used our proposed methods to evaluate the data set. First, we used four CNN models based on MFCC feature, which is trained on four folds to evaluate the data set. Second, we obtained the mean of the scores from four different models to ensure good performance. Third, we used eight models based on the MFCC and SIF features to evaluate the data set and obtain the mean of the scores for the final result.

5. CONCLUSION

In this report, we proposed two methods for the audio scene classification task. We presented a CNN model referred to as Inception-ResNet-v1, which was trained on the basis of MFCC features. Then, we analyzed the confusion matrix of the CNN models based on MFCC and SIF. Finally, we proposed a late-fusion method, which combined the scores of the trained CNN model based on the MFCC and SIF features, thereby improving the performance of the audio scene classification.

6. ACKNOWLEDGMENT

This work is funded by the National Key Research and Development Program of China (NO. 2016YFB0200401)

7. REFERENCES

[1] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, “DCASE 2016 acoustic scene classification using convolutional neural networks,” DCASE2016 Challenge, Tech. Rep., September 2016.

[2] 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,” DCASE2016 Challenge, Tech. Rep., September 2016.

[3] S. Christian, I. Sergey, V. Vincent, and A. A. Alexander, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017.

[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] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.