ADSC SUBMISSION FOR DCASE 2017: ACOUSTIC SCENE CLASSIFICATION USING DEEP RESIDUAL CONVOLUTIONAL NEURAL NETWORKS

Shengkui Zhao[1], *Thi Ngoc Tho Nguyen*[1], *Woon-Seng Gan*[2], *Douglas L. Jones*[1*],

[1] Advanced Digital Sciences Center, Singapore, {shengkui.zhao, tho.nguyen, jones }@adsc.com.sg
[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore,
{ewsgan}@ntu.edu.sg

ABSTRACT

This report describes our two submissions to the DCASE-2017 challenge for Task 1 (Acoustic scene classification). The first submission is motivated by the superior performance of the deep residual networks for both image and audio classifications. We propose a modified deep residual architecture trained on log-mel spectrogram patches in an end-to-end fashion for acoustic scene classification. We configure the number of layers and kernels for the deep residual nets and find that the modified deep residual net of 34 layers using binaural input features perform well on the DCASE-2017 development dataset. In the second submission, we implement a shallower network that consists of 3 convolutional layers and 2 fully connected layers to benchmark the performance of the residual network. Our two approaches improve the accuracy of the baseline by 10.8% and 10.6% respectively on the development set, and 9% and 6.9% on the unseen test set. We suggest that the size of the dataset for Task 1 is relatively small for deep networks to significantly outperform shallower ones.

Index Terms— acoustic scene classification, convolutional neural networks, deep learning, DCASE, ResNet

1. INTRODUCTION

The objective of acoustic scene classification is to classify a short audio record into one of the labeled classes. In the Detection and Classification of Acoustic Scene and Events (DCASE) 2017 challenge [1], a development dataset of 15 acoustic scenes with 312 10-second audio segments for each scene is provided. In both development and evaluation stages, the training data and the test data are ensured to include distinct recording locations for the same scene. Classifiers are to be developed using on only the training data and evaluated using on the provided test data.

The log-mel spectrogram that represents an audio segment as an image is so far the most frequently used input feature for audio classification. For the acoustic scene classification, the image-scene mapping preserves similar learning procedures as for the image-object mapping in the image classification. Many studies show that the capability of the deep models proposed originally for image classification is naturally extendable to the audio classification [2, 3, 4, 5]. Among the proposed deep models, Convolutional Neural Networks (CNNs) have been proven to be effective in the acoustic scene classification, and provide competitive performances compared to the other classification methods [6].

Driven by the increasing sizes of databases for image and audio classification, deeper CNN architectures become practical and attractive [7, 5]. Many studies show that network depth is of crucial importance and all the winners on the challenging ImageNet dataset employ very deep models [8, 9, 10, 11]. However, optimizing a deep CNN model is not a simple task. The study in [7] reveals that for a plain CNN network, when the network depth increases, both training and test accuracies get saturated and then degrade rapidly. Before 2015, researchers only successfully trained CNN models as deep as a few tens layers [12]. Fortunately, this problem has been addressed by using a deep residual learning framework that substantially eases the training of deep networks [7]. The basic idea is to reformulate the CNN layers as learning residual functions with reference to the layer inputs, in stead of learning original functions. Many experiments show that the nets with learning residuals are much easier to optimize than the counterpart ”plain” nets and enjoy accuracy gains from greatly increased depth [7, 5].

Considering the increased data size in DCASE-2017 challenge, we are interested to find out if deeper CNN models can improve the accuracy of the acoustic scene classification. Although many deep CNN models have been proposed for the acoustic scene classification in DCASE-2016 challenge, they usually have the number of convolutional layers less than 15. Motivated by the deep residual learning framework, we configure a deep residual net of 34 layers and exploit the accuracy gain by increased network depth. In our net configuration, all of the convolutional layers have 3×3 filter size and the input log-mel spectrogram patch has 128×128 dimensions. The kernel sizes are set accordingly for the data size. By varying the number of input channels, we find that using the binaural input channels enjoys accuracy gain compared to using only one monaural input channel. Our proposed deep residual CNN model outperforms the baseline by 10.8 % in accuracy in the DCASE-2017 development set and by 9% in the DCASE-2017 test set.

We compare the performances of the proposed deep residual network against a shallower network to evaluate the effectiveness of deep networks on the DCASE-2017 acoustic scene dataset. We implement a relatively shallow neural network proposed by the New York University [4], and evaluate its performance on Task 1. The network consists of 3 convolutional layers and 2 fully connected layers. The shallow network obtains competitive performance as the deep residual network. From this observation, we suggest that the size of the DCASE-2017 dataset for acoustic scene classification might not be sufficiently large enough for deep models to significantly outperform shallow ones.

[*]This study is supported by the research grant for the Human Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapores Agency for Science, Technology and Research (A*STAR).

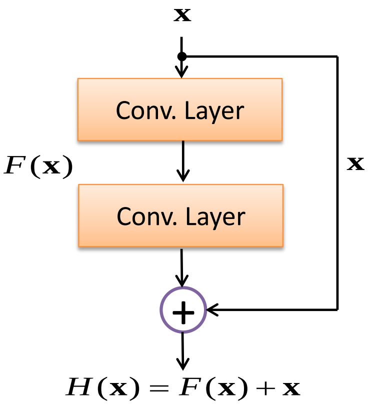

Figure 1: Residual building block unit.

2. DEEP RESIDUAL CONVOLUTIONAL NEURAL NETWORKS

2.1. Residual learning

A building block of the residual learning is shown in Fig. 1 [7]. The sub-blocks stand for the complete convolutional layers including the activation functions. A deep residual net can consist of many such building blocks by stacking them together. In one residual building block, the output $H(\mathbf{x})$ of the block is a mapping of the input $\mathbf{x}$. Instead of letting the multiple convolutional layers directly approximate the mapping $H(\mathbf{x})$, the residual mapping $F(\mathbf{x}) = H(\mathbf{x}) - \mathbf{x}$ is to be approximated. A shortcut connection from the input to the output adds an identity mapping to the output of the stacked layers. Identity shortcut connections add neither extra parameters nor computational complexity. Empirical evidences show that the residual networks are easier to optimize, and can improve classification accuracy from the considerably increased depth and data size. The entire network can still be trained by the back-propagation method.

2.2. The proposed residual net architecture

The proposed network architecture is a modified 34-layer residual net (MResNet-34) as shown in Fig. 2. In the MResNet-34, there are two types of residual building blocks: A and B which are shown in Fig. 3 and Fig. 4, respectively. The difference between block A and block B is the starting point of the shortcut connection. In block A, the shortcut connection starts after the batch normalization (BN) [10] and the rectified linear unit (ReLU) activation [13]. In block B, the shortcut connection starts directly from the input of the block. In Fig. 2, the numbers on the right side of block B indicate the number of times that block B of the same settings is repeated. All of the convolutional layers use 3 × 3 filter size. The stride, zero padding, and kernel size are given by the values of S, P, and K, respectively. Following the first convolutional layer, a 3 × 3 max pooling with a stride of 2 is applied. The size of the input patch is therefore reduced from 128 × 128 to 64 × 64 after the max pooling. At the end of the MResNet-34, a global average pooling is performed. A fully-connected layer with the softmax activation produces the class probabilities.

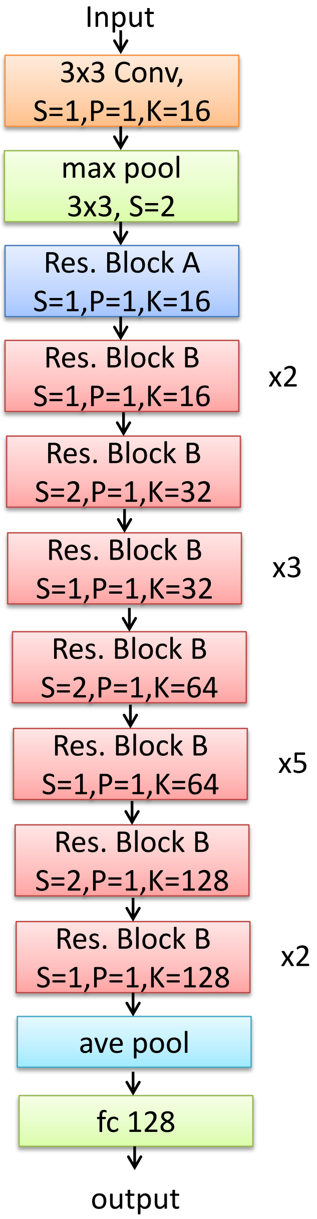

Figure 2: The proposed deep residual convolutional neural network architecture (MResNet-34) for the acoustic scene classification. The max pool and global average pool are not counted in the convolutional layers. There are many repeated block Bs with same settings as indicated by the number on the right side of the blocks to increase the depth. Much deeper networks can be configured by repeating block B several times.

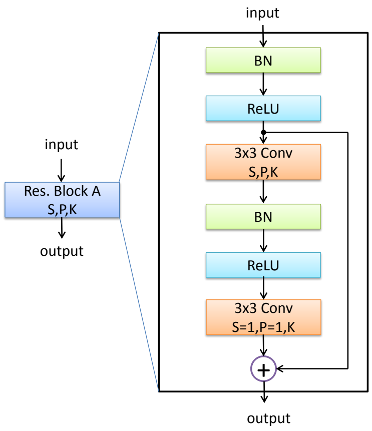

Figure 3: Residual building block A: the shortcut connection starts after ReLU.

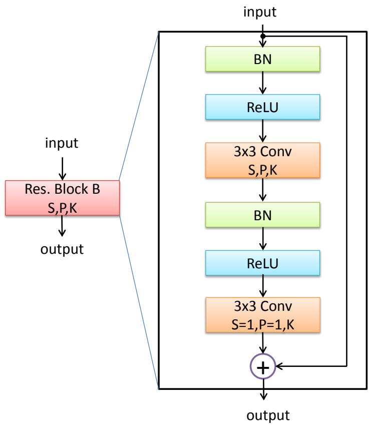

Figure 4: Residual building block B: the shortcut connection starts from the input of the block.

2.3. Feature representation and preprocessing

We choose to use the log-mel spectrogram patches as the input feature representation for our system. To extract log-mel spectrograms, we use the Librosa library [14] and set 128 frequency components for the audio signals sampled at 44.1 kHz. We use a Hamming window with a size of 46 ms (2048 samples at 44.1 kHz) with 50% overlap. The audio excerpts are first transformed into the frame-based mel-spectrograms, and then are taken log scales on the mel-spectrograms. For each evaluation, a global mean and a global standard deviation of each log-mel band are calculated from the corresponding training set. Then each band of the log-mel spectrograms of the training and test data is normalized by subtracting its global mean and dividing by its global standard deviation. We finally form non-overlap log-mel spectrogram patches of 128×128 (128 frames and 128 mel-spaced frequency bins) for all audio excerpts. Each log-mel spectrogram patch can be treated as a mono-channel image and the target label is given by the acoustic scene class. Considering the binaural data is provided in DCASE-2017 challenge, we are interested to investigate both the monaural setup and the binaural setup in our experiments. For the monaural setup, we extract log-mel spectrograms from only the averaged monaural channel. For the binaural setup, we extract log-mel spectrograms not only from the left and right channels, but also from the average and difference between the left and the right channels.

2.4. Model implementation and training

Our model was implemented using TensorFlow [15] and was trained using a single GPU. The parameters of our models are optimized with mini-batch stochastic gradient descent (SGD) with the momentum fixed at 0.9 throughout the entire training. The mini-batch size was set to 128 samples. The learning rate was initialized to be 0.02 and reduced to half every 10 epochs. We applied an $L2-$weight decay penalty of 0.0002 on all trainable parameters. Our training process took about 60 seconds for each epoch. For all the models, the training process ended after 100 epochs. To obtain the classification results at test stage, we first collected the individual class probabilities for each patch. We then averaged the probabilities of all patches from the same audio excerpt and assigned the class with maximum average probability.

Table 1: Accuracy of the monaural and binaural log-mel features.

Model	Feature	Fold 1	Fold 2	Fold3	Fold 4	Average
MResNet-34	Mono	81.2%	82.5%	78.8%	82.5%	81.3%
	Binaural	**86.0%**	**87.8%**	**82.6%**	86.0%	**85.6%**
NYU	Mono	83.5%	85.2%	79.9%	86.0%	83.6%
	Binaural	84.8%	87.1%	81.7%	**88.1%**	85.4%

3. A SHALLOW CONVOLUTIONAL NEURAL NETWORK

The neural network proposed by Salamon and Bello [4] consists of 3 convolutional layers and 2 fully connected layer. We refer to this model as NYU model. We slightly modify the network to speed up the convergence and reduce overfiting. We add batch normalization before the activation function in all 5 layers. We also add $L2-$regularization for all of the weights of the convolutional layers with a penalty factor of 0.001. The number of parameters of the NYU model is around $800,000$ while the number of parameters of the MResNet-34 is $1,300,000$. Even though the NYU model is much shallower than the MResNet-34, due to the last two fully connected layers and the higher number of filters in the convolutional layers, the number of parameters of the NYU model is still more than half of the number of parameters of the MresNet-34. We use the same input to train the MResNet-34 model to train the NYU network. The procedure to obtain the classification results is similar to those of the MResNet-34 model.

4. EVALUATION RESULTS

4.1. Dataset and metrics

We report the results of the proposed MResNet-34 model and the NYU model on the TUT DCASE-2017 development dataset. The dataset contains 4680 audio records of total 13 hours with 15 different indoor and outdoor acoustic scene classes: *beach, bus, cafe/restaurant, car, city center, forest path, grocery store, home, library, metro station, office, park, residential area, train and tram*. Our experiments are conducted using the 4-fold cross-validation setup provided by the DCASE-2017 challenge where in each fold, three-fourths of development data is used for training and one-fourth of development data is used for test.

Table 2: Class-wise accuracy of the MResNet-34 model using the binaural log-mel input features on the DCASE-2017 development dataset.

Scene	Fold 1	Fold 2	Fold 3	Fold 4	std.
Beach	82.1%	79.5%	98.7%	80.8%	7.8%
Bus	88.5%	93.6%	96.2%	82.1%	5.4%
Café/Rest.	65.4%	83.3%	74.4%	76.9%	6.4%
Car	98.7%	96.2%	96.2%	97.4%	1.1%
City center	88.5%	88.5%	93.6%	92.3%	2.3%
Forest path	94.9%	98.7%	89.7%	89.7%	3.8%
Grocery store	88.5%	97.4%	69.2%	76.9%	10.8%
Home	97.4%	97.5%	65.4%	70.5%	14.9%
Library	55.1%	74.4%	82.1%	82.1%	11.0%
Metro station	94.9%	100.0%	97.4%	100.0%	2.1%
Office	98.7%	98.7%	100.0%	98.7%	0.6%
Park	85.9%	88.5%	37.2%	64.1%	20.6%
Resident. Area	75.6%	76.9%	88.5%	84.6%	5.3%
Train	75.6%	78.2%	55.1%	93.6%	13.7%
Tram	100.0%	65.4%	96.2%	100.0%	14.5%

Table 3: Class-wise accuracy of the NYU model using the binaural log-mel input features on the DCASE-2017 development dataset.

Scene	Fold 1	Fold 2	Fold 3	Fold 4	std.
Beach	85.9%	64.1%	97.2%	84.4%	11.9%
Bus	94.1%	84.6%	91.8%	96.4%	4.4%
Café/Rest.	67.4%	91.0%	51.3%	76.4%	14.4%
Car	97.4%	93.6%	100.0%	100.0%	2.6%
City center	92.3%	95.4%	87.7%	88.5%	3.1%
Forest path	96.4%	98.7%	89.2%	98.7%	3.9%
Grocery store	88.2%	92.8%	76.9%	86.4%	5.8%
Home	89.5%	82.7%	75.6%	69.7%	7.4%
Library	45.4%	71.0%	82.8%	75.4%	14.1%
Metro station	94.1%	100.0%	100.0%	98.7%	2.4%
Office	100.0%	99.2%	92.8%	99.2%	2.9%
Park	88.2%	88.7%	50.5%	70.3%	15.7%
Resident. Area	76.7%	84.9%	90.8%	80.3%	5.3%
Train	55.6%	89.0%	44.6%	97.4%	22.1%
Tram	100.0%	70.5%	94.9%	99.7%	12.2%

In Table 1, we show the classification accuracy of the two models for each fold using the monaural and the binaural setup. It is observed that the binaural setup is better than the monaural setup for all folds. It is noted that the training patches in the binaural setup are 4 times of the training patches in the monaural setup. The increase of training patches boosts the model performance. The binaural setup has 4.3% higher in average accuracy using the MResNet-34 model. The gap is smaller using the NYU model.

In Table 2, we show the class-wise accuracy of the MResNet-34 model with the binaural setup for each fold. The class-wise accuracy varies across different classes and also varies across different folds. The standard deviations (std) of the class-wise accuracies of all four folds are varying from 0.6% to 20.6%. The *Office* has the smallest standard deviation while the *Park* has the largest standard deviation. It indicates that the model is affected by the ways of splitting the training data and the test data. The model learns well from all the splits of the *Office* data, while has learning challenge from the splits of the *Park* data. For the other classes, the model has learning capabilities in-between. Table 3 shows the class-wise accuracy of the NYU model. Similar observations can be made for the NYU model. In overall, the MResNet-34 has a mean of 8.02% in the standard deviations while the NYU model has a mean of 8.54%.

Table 4: The average class-wise accuracy over 4 cross-validation folds.

Scene	Baseline	MResNet-34	NYU
Beach	75.3%	**85.3%**	82.9%
Bus	71.8%	90.1%	**91.7%**
Café/Rest.	57.7%	**75.0%**	71.5%
Car	97.1%	97.1%	**97.8%**
City center	90.7%	90.7%	**91.0%**
Forest path	79.5%	93.3%	**95.8%**
Grocery store	58.7%	83.0%	**86.1%**
Home	68.6%	**82.7%**	79.4%
Library	57.1%	**73.4%**	68.7%
Metro station	91.7%	98.1%	**98.2%**
Office	**99.7%**	99.0%	97.8%
Park	70.2%	68.9%	**74.4%**
Resident. Area	64.1%	81.4%	**83.1%**
Train	58.0%	**75.6%**	71.7%
Tram	81.7%	90.4%	**91.3%**
Average	74.8%	**85.6%**	85.4%

Table 4 shows the average class-wise accuracies of the baseline, the MResNet-34, and the NYU model over all four cross-validation folds. The baseline model is a two-layer multi-layer perception (MLP) with 50 hidden units for each layer. Our proposed MResNet-34 model outperforms the baseline in 11 classes, and performs equally well in 2 classes, and performs slightly worse in 2 classes. The MResNet-34 is 10.8% higher than baseline and 0.2% higher than the NYU model.

4.2. Our submission to DCASE-2017 challenge

To obtain the evaluation results for submission to the DCASE-2017 challenge, we trained the MResNet-34 model and the NYU model using all the development data provided in the DCASE-2017 challenge. The input patches were retrieved from the binaural setup. The parameter settings and the training process are the same as what we made for the four-fold cross-validation. The MResNet-34 and the NYU model achieved 70% and 67.9%, respectively, on the final unlabelled evaluation data set which were 9% and 6.9% higher than the DCASE-2017 baseline model, respectively. The MResNet-34 is 2.1% higher than the NYU model.

5. CONCLUSIONS

We propose a deep CNN model with residual learning structure (MResNet-34) for acoustic scene classification. We observe that using the binaural setup has advantage over the monaural setup. It indicates that the model benefits from the increase of the training data size. The MResNet-34 model improves the baseline model by 10.8% and 9% more on development set and test set, respectively. We also show that a shallower network can also work well on the provided data size with higher accuracies on many class scenes than the deeper MResNet-34. It might suggest that the deeper models will gain more on the increase of data size and ensemble method of many models of different depth is more desired.

6. REFERENCES

[1] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, November 2017, submitted.

[2] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, September 2015.

[3] I. V. McLoughlin, H. min Zhang, Z.-P. Xie, Y. Song, and W. Xiao, "Robust sound event classification using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 540–552, 2017.

[4] J. Salamon and J. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 3 2017.

[5] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. W. Wilson, "CNN architectures for large-scale audio classification," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 131–135, 2017.

[6] 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," in *Detection and Classification of Acoustic Scenes and Events*, September 2016.

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

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *The International Conference on Learning Representations (ICLR)*, 2015.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *The International Conference on Computer Vision (ICCV)*, 2015.

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *The Thirty-second International Conference on Machine Learning (ICML)*, 2015.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

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

[13] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27 th International Conference on Machine Learning*, 2010.

[14] B. McFee, M. McVicar, O. Nieto, S. Balke, C. Thome, D. Liang, E. Battenberg, J. Moore, R. Bittner, R. Yamamoto, D. Ellis, F.-R. Stoter, D. Repetto, S. Waloschek, C. Carr, S. Kranzler, K. Choi, P. Viktorin, J. F. Santos, A. Holovaty, W. Pimenta, and H. Lee, "librosa 0.5.0," Feb. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.293021

[15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Man, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vigas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2015. [Online]. Available: http://download.tensorflow.org/paper/whitepaper2015.pdf