ACOUSTIC SCENE CLASSIFICATION USING AUTOENCODER

Xiaoshuo Xu, Xiaoou Chen, Deshun Yang

Peking University, Beijing, China, {xsxu, chenxiaoou, yangdeshun}@pku.edu.cn

ABSTRACT

This report describes our contribution to the Acoustic Scene Classification (ASC) task of the 2017 IEEE AASP DCASE challenge[1]. We apply an Autoencoder to capture the discriminative information underlying the audio. Then, a Logistic Regression model is employed to recognize different scenes under the compressed representation. In order to boost the performance, we train models based on different channels from the original recordings and simply apply majority voting method on the predictions. Our final system achieves 84.31% on a four-fold cross-validation setting, which outperforms the baseline system by 9.5%.

Index Terms— Acoustic Scene Classification, Autoencoder, Logistic Regression, Ensemble

1. INTRODUCTION

ASC is defined as automatically identifying the acoustic context from the recordings. Prevalent features include some speech inspired features like log mel-band energy[1] and MFCC[2, 3]. Besides, some feature learning algorithms like NMF and Sparse Coding are applied to learn intermediate and sparse representation[4, 5]. Then, classifiers like Support Vector Machine (SVM), Gaussian Mixture Model (GMM) and Neural Network (NN) are used to predict the scenes based on the features. Furthermore, ensemble techniques are utilized to boost the identification.

In this paper, we follow these steps and develop our own approach. We propose an Autoencoder model to learn a representation of the data. Then, a Logistic Regression model is employed to recognize different scenes under the compressed representation. Combining training steps of Autoencoder and Logistic Regression, our model captures more discriminative information and achieves better performances than training the two models separately. In addition, we fuse the models trained from different channels of the original audio and obtain a more robust model.

2. APPROACH

2.1. Constant-Q Transform

In our experiments, we apply Constant-Q Transform (CQT) to the audio (we use *Yaafe*[6] to extract this feature), as previous research has successfully applied in this task[5, 7]. The feature is extracted from the raw audio with 24 bins per octave from 98Hz to 22050Hz. Besides, the CQT kernels are aligned in the center of each frame. As the sample rate of the original recordings is 44100Hz and the hop size is set to 4096, the resulting feature of each recording could be written as a matrix with the size of 108 x 188, where 108 is the number of time frames and each frame has 188 frequency bands. To accelerate the training step of Neural Network, we normalize the spectrograms among the training set. For exploiting more temporal information of data, we concatenate five consecutive spectrograms into a vector with the hop size of one in each recording. Hence, we obtain a matrix with the size of 104 x 940.

2.2. Autoencoder

Autoencoder is a kind of Neural Network used to learn a representation of a set of data, especially for the purpose of dimensionality reduction. The reason we employ this technique is that we hope to generate low-dimensional and discriminative features from CQT instead of applying a classifier directly to the spectrograms. In our model, the structure of the network is simple and contains three layers. The hidden layer includes 256 nodes with relu activation, and no activation is used on the output layer. Given the concatenate spectrogram $x \in R^{940}$, the output of the middle layer can be expressed as follows:

$$h = max(W_1 x + b_1, 0) \tag{1}$$

where h denotes the output of the hidden layer. Finally, the square reconstruction loss is used to train the Autoencoder model.

$$L_1 = \sum_n ||x_n - (W_2 h_n + b_2)||^2 \tag{2}$$

2.3. Logistic Regression

A multinomial logistic regression is applied to learn a classifier. Given the compressed feature h of a frame, the posterior probability of a acoustic scene C_k is given by a softmax transformation of linear functions of h, so that

$$y_k = P(C_k|h) = \frac{exp(w_{3,k}^T h + b_{3,k})}{\sum_j exp(w_{3,j}^T h + b_{3,j})} \tag{3}$$

where both $W_3 = (w_{3,1} \dots w_{3,M})^T$ and $b_3 = (b_{3,1} \dots b_{3,M})^T$ are parameters of the model (M is the number of category and equal to 15 in this case). Furthermore, the loss function for the classification problem could be defined as:

$$L_2 = -\sum_{n,k} t_{nk} y_{nk} \tag{4}$$

where $t_n = (t_{n1}, t_{n2} \dots t_{nk})^T$ represents the label of the category and $y_n = (y_{n1}, y_{n2} \dots y_{nk})^T$ indicates the predicted probabilities for each class. In the experiments, we consider the concatenate spectrograms in each recording as training samples and assign the label of the recording to all spectrograms. Then, when testing the data, the system sums up the predicted probabilities among the frames in each audio. The category with the maximum is the predicted acoustic scene.

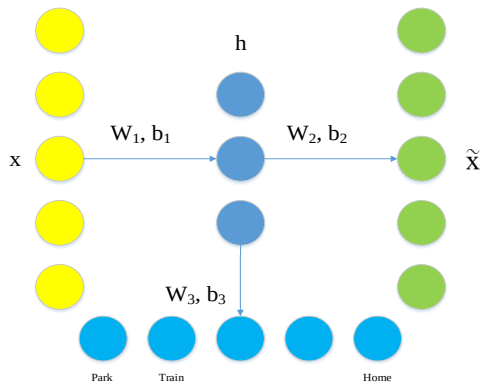


Figure 1: Network Structure

2.4. Combination of training procudures

The aforementioned methods result in a model that could be used to identify the acoustic scene. However, instead of training the Autoencoder and Logistic Regression model separately, we try to combine the learning steps of Autoencoder and Logistic Regression together, see Figure 1. That is, the network has two outputs, and the gradients of them are both used in backpropagation. Experimental results show that in this way, the model could learn discriminative representations and achieves better performance than implementing the models separately, see Section 3.1. The global loss function L takes the reconstruction loss and the classification loss into consideration, as follows:

$$L = L_1 + \alpha L_2 \tag{5}$$

where α is a hyperparameter weighting the importance of the Autoencoder model. When α is equal to zero, the model degrades into a simple natural network, very similar to the model used in the baseline.

The model is implemented in Keras[1]. We apply the Adam optimization algorithm to the model with the learning rate of 0.001 and train the model for four epoches. As the features are normalized in the preprocessing, we find that several training epoches are enough to guarantee the convergence.

2.5. Model ensemble

Audio material often contains two channels of tracks, so do the recordings in the task. In our experiments, we extract CQT features from the channels respectively and the average channel. Then, for each channel, a neural network is trained to fit the data. Finally, the models are fused through voting method, i.e. the predicted probabilities of the models are summed up, and the prediction is the scene with the highest score. One might note that such voting method is simple, and perhaps more sophisticated algorithm might help to achieve better results. However, in our experiments, we implement some other ensemble methods, but fail to obtain a stronger model than it.

We admit that our ideas are motivated by [8], which employed the similar technique and won the 1st place in the Acoustic Scene Classification task of DCASE 2016. Since it is very possible that some subtle acoustic signals are only captured by one channel, converting stereo audio into mono might lose lots of information and result in poor identification. Besides, we think that voting method is a good choice for fusing models, because it's natural to assume that different channels contribute to recognition equally. On the contrary, modeling their differences with complicated ensemble methods might cause serious overfitting.

[1] https://keras.io/

Method	$\alpha = 0$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.5$
Precision	0.7678	0.7702	0.7717	0.7779

Method	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	Seperation
Precision	0.7661	0.7796	**0.7943**	0.7500

Table 1: Influence of α on the model. The mean precision is computed on the four-fold cross-validation setting. Seperation represents training Autoencoder and Logistic Regression separately.

Model	L-M	R-M	M-M	LR-M	LRM-M
Precision	0.7783	0.7892	0.7943	0.8361	**0.8431**

Table 2: Comparision of different fusing models. L-M, R-M and M-M represents the models trained from left channel, right channel and mean channel respectively. LR-M is the fusing model of L-M and R-M, and LRM-M is attained from L-M, R-M and M-M.

3. EXPERIMENTS

In this section, we design several experiments to verify the validity of our model. First, we investigate the influence of α on the result. Then, we combine the models trained from different channels and compare their performances. Finally, the experimental results and the submitted systems are presented.

3.1. Tuning the model

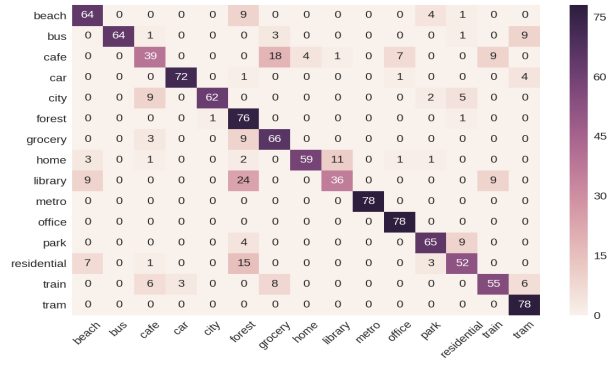
Fixed the hyperparameters like learning rate, we iterate α through $\{0, 0.2, 0.3, 0.5, 0.7, 0.8, 0.9\}$ and train the model. As shown in Table 1, with the increase of α, the average precision shows a trend of growth (Note that when $\alpha = 0$, the model is exactly the same as a simple neural network). Especially, when $\alpha = 0.9$, the precision achieves 79.43%. Based on the observation, we set $\alpha = 0.9$ in the following experiments.

Besides, we also train the Autoencoder model and Logistic Regression model separately for comparision purposes. As shown in Table 1, combining the two models indeed outperforms utilizing the models separatively. Such improvement might be due to the aggregation of the supervised learning and unsupervised learning. The supervised labels adapt the Autoencoder model to exploit information lying in data, and in the same time the intermediate representation contributes to learn a good classifier. We will apply this algorithm on other classification tasks and study this model thoroughly in the future.

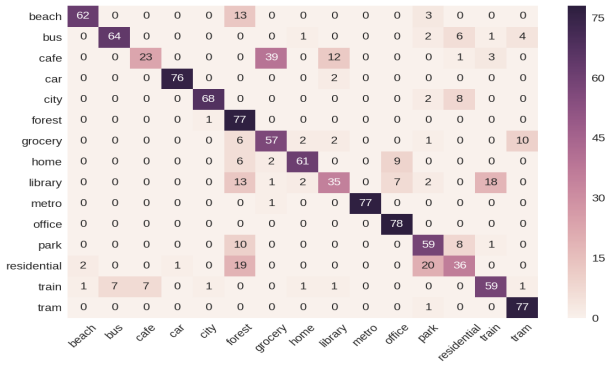
3.2. Fusing the models

Then, we combine the models trained from different audio channels and compare their performances. Table 2 shows that mixing the models helps to improve the prediction. LR-M surpasses the simple models L-M, M-M and R-M, and especially fusing the three

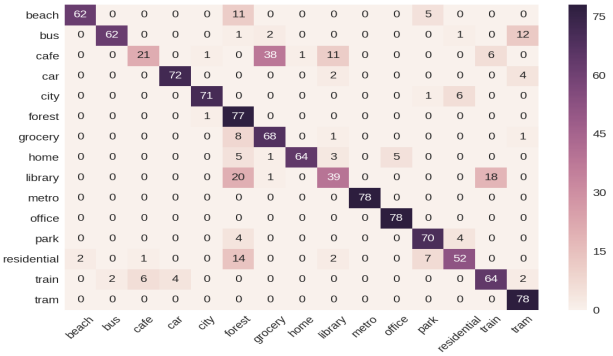
models, LRM-M outperforms L-M, R-M and M-M by about 6% and achieves the highest accuracy among them. Such improvement is indeed promising and encouraging.



(a) L-M



(b) R-M



(c) LR-M

Figure 2: Confusion matrixes of different models on Fold 1. The vertical axis represents the ground truth labels, and the horizontal axis indicates the predicted labels.

To learn the advantage of fusing models, we draw the confusion matrixes of different models, see Figure 2. It is very interesting to find that, even though the simple models achieve similar precisions, they perform differently on specific acoustic scenes. For instance, L-M obtains higher accuracy on residential area than that of R-M. Meanwhile, L-M tends to wrongly recognize forest path as library, while R-M shows no such strong bias. Taking advantage of different models, the fusing model LR-M reduces the bias to some extent and achieves great performance; as shown in Figure 2 (c), some misclassified samples are rectified and the diagonal line becomes darker. Especially, regarding to park and residential area, the fusing model LR-M performs better than L-M and R-M. One may note that for some categories like car, LR-M actually becomes a bit worse. However, it turns out to be a better model overall.

	Mean	Fold 1	Fold 2	Fold 3	Fold 4
Precision	**0.8431**	0.8342	0.8764	0.8338	0.8282

Table 3: Final result

3.3. Final submission

Table 3 shows the results of our approach on the development set. Our model achieves the average precision of 84.31%, outperforming the baseline system by 9.5%. Based on this model, we submit three results to the challenge. The first one is the prediction of the model trained on Fold 2, since it achieves the highest accuracy among the four folds[2]. The second one averages the prediction of the four folds[3]. That is, sum up the predicted possibilities for each scene and output the category with the maximum. The last one is the prediction of the model trained from the development set[4].

4. CONCLUSION

We have presented our system to the ASC task of 2017 IEEE AASP DCASE challenge. Our approach combines Autoencoder and Logistic Regression and obtains a robust model. In addition, fusing the models trained from different channels, our model exploits significant information underlying the audio and achieves good results.

[2]XU_PKU_task1_1
[3]XU_PKU_task1_2
[4]XU_PKU_task1_3

5. 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] J.-J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *The Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.

[3] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[4] J. Nam, Z. Hyung, and K. Lee, "Acoustic scene classification using sparse feature learning and selective max-pooling by event detection," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[5] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Acoustic scene classification with matrix factorization for unsupervised feature learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 6445–6449.

[6] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "Yaafe, an easy to use and efficient audio feature extraction software." in *ISMIR*, 2010, pp. 441–446.

[7] V. Bisot, S. Essid, and G. Richard, "Hog and subband power distribution image features for acoustic scene classification," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 719–723.

[8] 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.