Acoustic Scene Classification Using Deep Mixtures Of Pre-trained Convolutional Neural Networks

Truc Nguyen, Alexander Fuchs and Franz Pernkopf
Signal Processing and Speech Communication Laboratory
Graz University of Technology
Graz, Austria/Europe
{t.k.nguyen, fuchs, pernkopf}@tugraz.at

***Abstract*—We propose a heterogeneous system of Deep Mixture of Experts (DMoEs) models using different Convolutional Neural Networks (CNNs) for acoustic scene classification (ASC). Each DMoEs module is a mixture of different parallel CNN structures weighted by a gating network. All CNNs use the same input data. The CNN architectures play the role of experts extracting a variety of features. The experts are pre-trained, and kept fixed (frozen) for the DMoEs model. The DMoEs is post-trained by optimizing weights of the gating network, which estimates the contribution of the experts in the mixture. In order to enhance the performance, we use an ensemble of three DMoEs modules each with different pairs of inputs and individual CNN models. The input pairs are spectrogram combinations of binaural audio and mono audio as well as their pre-processed variations using harmonic-percussive source separation (HPSS) and nearest neighbor filters (NNFs). The classification result of the proposed system is 72.1% improving the baseline by around 12% (absolute) on the development data of DCASE 2018 challenge task 1A.**

***Index Terms*—Acoustic scene classification, convolutional neural network, mixture of experts, nearest neighbor filters.**

I. Introduction

Acoustic scene classification (ASC) is a recognition task for sounds of environments called acoustic scenes. These scenes are assumed to be distinguishable from other scenes based on its acoustic properties. They are a characterization of a location or situation. An acoustic scene is composed of sound events which are considered as important descriptors. However, in real environments, these sound events are varying and can have different degrees of overlap. Therefore the acoustic scenes are unstructured and often unpredictable in its occurrence causing more challenges for ASC compared to speech and music signal processing.

ASC includes two stages contributing to its effectiveness; namely are feature extraction and classification. Mel-frequency scales such as mel-frequency cepstral coefficients (MFCCs) and log-mel energies have been the most popular features applied in ASC. Furthermore, gamma-tone filter spectrogram, constant-Q transform spectrogram, waveform and scalogram features have been used [1], [2], [3], [4], [5]. In addition, the features can be used as basis for higher level features i.e., i-vectors [6], typical features of image processing i.e. histogram of gradients (HOGs) [7], local binary pattern (LBP) [8] and learned features, [9], [10], [11], [12], [13], [14].

In the recent ASC, deep learning is the method of choice for classification. For example, a variety of model structures have been applied such as parallel CNNs of VGGNets [9], Xception networks [15], DenseNets [16], or CliqueNets [17]. Furthermore, ensemble methods have been a key factor contributing to successfully proposed ASC systems. Popular ensemble methods such as averaging, weight averaging ensemble [11], [18], ensemble selection [9], [12], or random forests [10] have been used for different models. Recently, snapshot averaging hase been proposed. It is an ensemble method which allows to provide many different models with only one training run using cyclical cosine learning rate schedule [13]. Moreover, popular data augmentation techniques i.e., Generative Adversarial Networks (GANs) [4], [19], data mixup [9], [17], SpecAugment [18] and transfer learning [1], [3] are used to improve performance.

In this paper, we use a mixture of experts. The experts use pre-trained CNN models and embed them into the DMoEs model. The expert outputs are combined using corresponding weights of the gating network. The gating network, which is similar to an attention mechanism, is a deep neural network located inside of the DMoEs model. It benefits from using the same input information as the pre-trained experts. Furthermore, we use different features for DMoEs, i.e. either using log-mel spectrogram of both channels and their pre-processed variations such as harmonic-percussive source separation and nearest neighbor filters (NNFs). These features have been also used in the best sytems of DCASE 2017 and DCASE 2018 challenges [9], [10], [12]. In addition, mixup data augmentation and ensemble techniques are used to enhance the model performance.

The rest of the paper is organized as follows. Section 2 presents the proposed ASC system, including audio processing, NNF features, mixup data augmentation, parallel CNNs, deep mixture of experts and ensemble methods. In Section 3, we provide experiments and evaluate the performance of the proposed approach. Section 4 concludes the paper.

II. Proposed system

The proposed system is illustrated in Fig.1. The system consists of three stages. First, the binaural and mono audio signals are converted to various time-frequency representations

978-1-7281-4550-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMLA.2019.00151

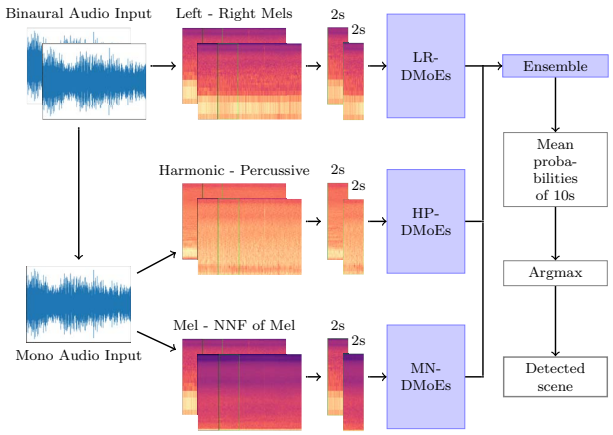

Fig. 1. Proposed System.

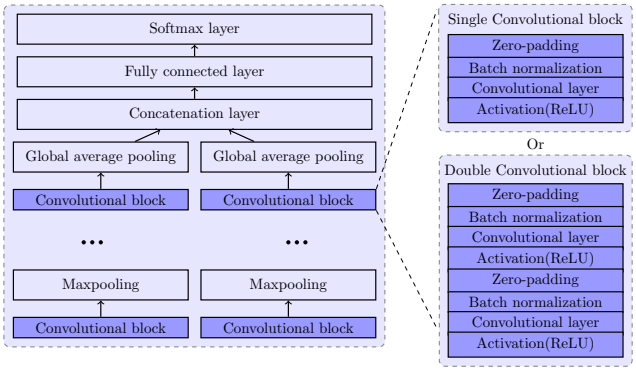

Fig. 2. Parallel CNNs with single and double convolutional blocks.

chunked into 2s segments. These features are used for training the CNN models and DMoEs. Finally, the probability outputs of these DMoEs models are fed to another ensemble method before making the final label predictions.

A. Audio Pre-processing

We use the DCASE 2018 data set which is recorded by binaural microphones at sampling rate of 48kHz. We keep the sampling rate and use both the left and right channels of the audio to exploit its richer spatial information. We extract 128 bin mel-energies of the binaural and mono (i.e. mean of binaural channels) channels. This is an appropriate number of bins compared to 40, 64 or 256 bins in [20], [21], [18] for representing the spectral characteristics. The window function of the short-time Fourier transform (STFT) is a Hann window and the window size is selected as 40ms with 20ms hop size. The size of each 2s segment is 128 bins x 100 frames.

We use left- and right- channel mel-spectrograms (LR), harmonic and percussive spectrograms (HP), and mono-mel spectrogram and its nearest neighbor filtered version (MN)[1]. Although in the best systems spectrogram splitting of 1s without overlap is used, splitting to 2s segments leads to a better performance in our case. All features are converted into logarithmic scale and normalized to zero mean and unit variance.

B. Nearest Neighbor Filter

Environmental sounds are often unstructured, neither predictable repetitions nor harmonic sounds that are composed by potentially overlapping sound events. These sound events could be periodic or randomly repeating sounds such as sounds of a siren, horn of vehicles, sounds of opening and closing metro doors at metro stations etc. Therefore, it is useful for an ASC system to generate features which emphasize the appearance of similar patterns of a sound event in an acoustic scene [12].

[1]The processing is done by using Librosa toolbox https://librosa.github.io/librosa/

Nearest neighbor filters are based on Repeating Pattern Extraction Technique (REPET) [22] for cases where repetitions happen intermittently or without a fixed period. The algorithm determines first the five most similar spectrogram frames as nearest neighbors by using a similarity matrix. Then the median of the nearest neighbors is used to create a new spectrogram representation.

C. Mixup Data Augmentation

Mixup data augmentation [23] constructs virtual training examples ($\tilde{x}$, $\tilde{y}$) by a convex combination of two randomly selected training data samples (x_i, y_i) and (x_j, y_j), i.e.

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda) x_j, \\ \tilde{y} &= \lambda y_i + (1-\lambda) y_j, \end{aligned} \tag{1}$$

where x_i and x_j are the features of the 2s segments and y_i and y_j are the one-hot encoded class labels. $\lambda \in [0, 1]$ is acquired by sampling from a beta distribution *Beta*(α, α) with α being a hyper-parameter. We use α of 0.2 for our system.

D. Parallel Convolutional Neural Networks

Recently, CNNs have been considered as an extractor of high-level features. In this paper, we generate six CNN structures for each DMoEs model. By adjusting the depth of the CNNs and the structure of the convolutional blocks using a different number of single convolutional blocks and double convolutional blocks. It means that each CNN structure is considered as an expert and it can focus on certain parts of the acoustic scene.

Parallel CNNs (also called multi-input CNNs) have been used with different input features or structures for each branch of the CNN architecture. We feed different feature representations (LR, HP, MN) to parallel branches of CNNs. Subsequently, both branches are concatenated before the fully-connected layer. A convolutional block consists of zero-padding, batch normalization (BN), convolution layers (Conv) and followed by Rectifier Linear Units (ReLUs) activation function. Fig. 2 shows the structure of parallel CNNs using single and double convolutional blocks.

Based on empirical results, we select the number of filters for the convolutional layers of the CNNs including 2, 3 and 4 single or double convolutional blocks as 32 - 256, 32 - 128 - 256 and 32 - 64 - 128 - 256, respectively. Both convolutional layers of each double convolutional block have the same size of 3x3 filters.

E. Deep Mixture of Pre-trained Experts

Mixture of experts architecture (MoEs) consists of a set of modules referred to as expert networks which are appropriate for different regions of the input space. A gating network identifies the suitable expert for each regions [24]. When experts and the gating network are implemented by DNNs, it is called a Deep Mixture of Experts (DMoEs) [25], [26]. A DMoEs model needs a two-step training process, namely, learning of parameters for the individual experts and learning of the parameters for the gating network. Firstly, the experts are pre-trained for modeling the outputs y_i. These outputs are then combined by a set of weights determined by the gating network g_i. The combination of the pre-trained experts and their corresponding weights is the output of the DMoEs model. By post-training, the weights of the gating network are adjusted using the same input data as the experts. Fig.3(a) shows the architecture of a DMoEs. The output layer provides:

$$p(y|x;\theta) = \sum_{i=1}^{n} p(g=i|x;\theta_g)p(y|g=i,x;\theta_i) \qquad (2)$$

where θ_g are the parameters of the DNN gating network, θ_i are the parameters of the *i*-th expert and *n* is the number of experts.

In this work, each DMoEs model includes 6 different pre-trained CNNs, a gating network and an output layer. The gating network is implemented by either a parallel CNN or multi-perceptron layer. Its outputs are concatenated and fed to a softmax layer of *M* units, where *M* is equal to the product of the number of classes and the number of experts.

We use the same training dataset of the DCASE development set for both pre-training and post-training.

F. Ensemble Methods

We combine the DMoEs models using ensembling techniques. In particular, we compare performance of three ensemble methods named average ensemble (AE), weighted averaging ensemble (WE) and ensemble selection with replacement (ES) [27]. The weights of the average ensemble are equal for each DMoEs model and sum to one. The weighted averaging ensemble determines the optimal weights by minimization of the cross-entropy loss using ground-truth labels and estimated labels. Weights are also constrained to sum to one. Ensemble selection with replacement [27] is an iterative method that allows models to be added to an ensemble multiple times such that the performance of the combination is maximized, so the weights of this ensemble are equivalent to the number of times the model has been selected divided by the total number of models in the ensemble.

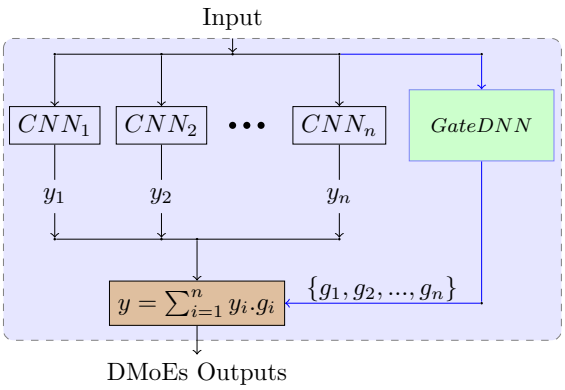

Fig. 3. Architecture of DMoEs.

We use the training data to determine the optimal weights for WE and ES. Sequential Least Squares Programming (SLSQP) is used to determine the weights of WE. For ES, we start with the best model among all candidate models of the ensemble before greedy model selection is performed for 200 iterations. The determined weights are used for evaluation on the test set.

III. Experiments

A. Data

We use the TUT Urban Acoustic Scene 2018 dataset recorded in six European cities. A binaural microphone at sampling rating 48kHz is used. The original recordings were split into 10-second segments that are provided in the individual files. The dataset includes 8640 segments with 6122 segments for training and 2518 segments for testing. It contains 10 scenes.

B. Setup

The validation set accounts for approximately 30% of the original training data and there are no segments from the same location and city in both training and validation data sets. Acoustic features are log mel-band energies of 128 frequency bands and their variations with 40 ms frame size and 50% hop size. The network training is carried out by optimizing the categorical cross-entropy using the stochastic gradient decent optimizer at a learning rate of 0.001. We use Glorot uniform data to initialize the network weights. The number of epochs and batch size was 500 and 128, respectively, and data is shuffled between epochs. Model performance is evaluated on the validation set after each epoch and the selected model is the best performing one on the validation set.

C. Performance On The Test Set

Table 1 presents the accuracy of the best individual models. The best CNN structures for each input combination are: LR using 4 single CNN blocks (LR-4s-cnn), HP using 3 double

TABLE I
ACCURACY OF THE PROPOSED MODELS AND OF THE ENSEMBLE METHODS.

Algorithm	Accuracy	Algorithm	Accuracy
Baseline [20]	59.7 ± 0.7	LR-4s-cnn(*)	67.6
-	-	HP-3db-cnn(*)	68.0
-	-	MN-2s-cnn(*)	67.5
LR-MoE_dnn	67.1	MN-MoE_cnn	67.4
HP-MoE_dnn	67.6	LR-MoE_cnn	67.5
MN-MoE_dnn	67.4	HP-MoE_cnn	66.6
AE(MoE_dnn)	70.9	AE(MoE_cnn)	70.2
WE(MoE_dnn)	70.0	WE(MoE_cnn)	70.2
ES(MoE_dnn)	71.0	ES(MoE_cnn)	70.5
AE-6(MoE_dnn + 3(*))	71.8	AE-21	**72.1**
WE-6(MoE_dnn + 3(*))	71.8	WE-21	71.3
ES-6(MoE_dnn + 3(*))	71.0	ES-21	71.2

CNN blocks (HP-3db-cnn) and MN using 2 single CNN blocks (MN-2s-cnn). We can see that the best models reach a comparable accuracy. Furthermore, MoE_dnn and MoE_cnn are DMoEs of 6 experts using different CNN structures and a gating network of either a multi-perceptron layer or convolutional layer, respectively. The performances of MoE_dnn and MoE_cnn with different inputs (LR, HP, MN) are slightly lower compared to the best corresponding models. The high variance of the gating weights for the 10 classes and 6 experts could partly cause the performance decrease.

In addition, Table 1 show performances of ensemble methods such as average ensemble (AE), weighted ensemble (WE) and ensemble selection (ES). Most of the performances of MoE_dnn ensembles using MoE_dnn models of 3 different inputs (LR, HP, MN) (called 3 MoE_dnn models) are higher than that of MoE_cnn ensembles. ES(MoE_dnn) is the best ensemble of 3 MoE_dnn models with a performance of 71.0%. The performance of the ensemble model can be improved by increasing the number of component models. Moreover, the ensemble performance of 6 individual component models including the 3 MoE-dnn models and the 3 best individual models (MoE_dnn + 3(*)) are a bit lower than the ensemble performance of 21 models which includes 6 different CNN structures and the MoE_dnn model for each of the 3 feature inputs (LR, HP, MN). The average ensemble of 21 model (AE-21) achieves the best performance of 72.1%.

Table 2 shows the class-wise accuracy of the baseline system and the proposed system. This system is the average ensemble of 21 models consisting of 18 individual models of different inputs and 3 DMoEs of DNN (MoE_dnn). The public square and the street traffic are the hardest and easiest scenes for both the proposed algorithm and baseline system, respectively. The proposed algorithm for the hardest scene significantly outperforms that of the baseline system by 14.2%.

IV. CONCLUSION

In this paper, we introduce a deep mixture of experts model to exploit the diversity of high-level features from pre-trained parallel CNNs. We combine these models using weights of a gating network. In order to improve the prediction performance, we propose an heterogeneous ensemble of 3 DMoEs models using a multi-layer perceptron in the gating network and 18 individual models of 6 different CNNs and 3 different audio features. Mixup data augmentation is additionally used to leverage the model accuracy. Our proposed ensemble system significantly improves performance to 72.1% and it outperforms the baseline system of the DCASE 2018 task 1A by 12% (absolute). However, it requires to train many models with a large number of parameters.

TABLE II
CLASS-WISE ACCURACY OF THE PROPOSED SYSTEM ON THE TEST SET COMPARED TO BASELINE SYSTEM.

Scene label	Baseline [20]	Propose
Airport	72.9	74.0
Bus	62.9	71.1
Metro	51.2	69.0
Metro station	55.4	83.4
Park	79.1	88.0
Public square	40.4	54.6
Shopping mall	49.6	54.8
Street_pedestrian	50.0	58.3
Street_traffic	80.5	92.7
Tram	55.1	74.7
Average	59.7 ± 0.7	**72.1**

ACKNOWLEDGMENT

This research was supported by Vietnamese - Austrian Government scholarship and by the Austrian Science Fund (FWF) under the project number I2706-N31. We acknowledge NVIDIA for providing GPU computing resources.

REFERENCES

[1] H. Phan R. Palaniappan L. Pham, I. McLoughlin and Y. Lang, "Bag-of-features models based on c-dnn network for acoustic scene classification," in *Audio Engineering Society Conference AES-International Conference on Audio Forensics*, 2019.

[2] A. Atul Agrawal P. Tilak and V. Ramasubramanian, "Acoustic scene classification using deep cnn raw-waveform," Tech. Rep., DCASE 2018 Challenge, 2018.

[3] Zhao Ren, Kun Qian, Yebin Wang, Zixing Zhang, Vedhas Pandit, Alice Baird, and Bjorn Schuller, "Deep scalogram representations for acoustic scene classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 662–669, 2018.

[4] Hangting Chen, Zuozhen Liu, Zongming Liu, Pengyuan Zhang, and Yonghong Yan, "Integrating the data augmentation scheme with various classifiers for acoustic scene modeling," Tech. Rep., DCASE2019 Challenge, June 2019.

[5] S. Hyeji and P. Jihwan, "Acoustic scene classification using various pre-processed features and convolutional neural networks," Tech. Rep., DCASE2019 Challenge, June 2019.

[6] M. Dorfer F Koreniowski K. Koutini B. Lehner, H. Eghbal-zadeh and G. Widmer, "Classifying short acoustic scenes with i-vectors and cnns: Challenges and optimisations for the 2017 dcase asc task," Tech. Rep., DCASE 2017 Challenge, 2017.

[7] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*.

[8] S. Abidin, R. Togneri, and F. Sohel, "Spectrotemporal analysis using local binary pattern variants for acoustic scene classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2112–2121, Nov 2018.

[9] Y. Han, J. Park, and Kyogu Lee, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," in *Proceedings of the DCASE 2016 Workshop (DCASE2016)*, 2016, pp. 11–15.
[10] Y. Sakashitaand M. Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," Tech. Rep., DCASE 2018 Challenge, 2018.
[11] H. Eghbal-zadeh H. Christop P. Fabian M. Dorfer, B. Lehner and W. Gerhard, "Acoustic scene classification with fully convolutional neural networks and I-vectors," Tech. Rep., DCASE 2018 Challenge, 2018.
[12] T.Nguyen and F. Pernkopf, "Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2018.
[13] H. Eghbal-zadeh K. Koutini and G. Widmer, "Acoustic scene classification and audio tagging with receptive-field-regularized CNNs," Tech. Rep., DCASE2019 Challenge, June 2019.
[14] Huy Phan, Oliver Y Chén, Lam Pham, Philipp Koch, Maarten De Vos, Ian McLoughlin, and Alfred Mertins, "Spatio-temporal attention pooling for audio scene classification," *Proc. Interspeech 2019*, 2019.
[15] C. Xinxing Y. Liping and T. Lianjie, "Acoustic scene classification using multi-scale features," Tech. Rep., DCASE2018 Challenge, September 2018.
[16] Dezhi Wang, Lilun Zhang, Kele Xu, and Yongxian Wang, "Acoustic scene classification based on dense convolutional networks incorporating multi-channel features," in *Journal of Physics: Conference Series*. IOP Publishing, 2019, vol. 1169, p. 012037.
[17] T. Nguyen and F. Pernkopf, "Acoustic scene classification with mismatched devices using cliquenets and mixup data augmentation," *Proc. Interspeech 2019*, pp. 2330–2334, 2019.
[18] Micha Komider, "Calibrating neural networks for secondary recording devices," Tech. Rep., DCASE2019 Challenge, June 2019.
[19] S. Mun, S. Park, D. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using svm hyper-plane," *Proceedings of DCASE2017 Workshop*, pp. 93–97, 2017.
[20] T. Heittola A. Mesaros and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2018.
[21] T. Iqbal-W. Wang Q. Kong, Y. Cao and M.D. Plumbley, "Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems," Tech. Rep., DCASE2019 Challenge, June 2019.
[22] Z. Rafii and B. Pardo, "Music/voice separation using the similarity matrix.," in *ISMIR*, 2012, pp. 583–588.
[23] H. Zhang, M. Cisse, Y. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
[24] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.
[25] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *Proceedings of the ICLR*, 2017.
[26] R. Salakhutdinov Z. Yang, Z. Dai and W. Cohen, "Breaking the softmax bottleneck: A high-rank rnn language model," *Proceedings of the ICLR*.
[27] R. Caruana, A. Niculescu-Mizel, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 18.