ACOUSTIC SCENE CLASSIFICATION USING A CONVOLUTIONAL NEURAL NETWORK ENSEMBLE AND NEAREST NEIGHBOR FILTERS

Technical Report

*Truc Nguyen**, *Franz Pernkopf*†

Graz University of Technology,
Signal Processing and Speech Communication Lab.,
Inffeldgasse 16c, A-8010 Graz, Austria/Europe,
{t.k.nguyen, pernkopf}@tugraz.at

ABSTRACT

This paper proposes Convolutional Neural Network (CNN) ensembles for acoustic scene classification of subtasks 1A and 1B of DCASE 2018 challenge. We introduce a nearest neighbor filter applied on spectrogram, which allows to emphasize and smooth similar patterns of sound events in a scene. We also propose a variety of CNN models for single-input (SI) and multi-input (MI) channels and three different methods for building a network ensemble. The experimental results show that for subtask 1A the combination of the MI-CNN structures using both of log-mel features and their nearest neighbor filtering is slightly more effective than that of single-input channel CNN models using log-mel features only. This statement is opposite for subtask 1B. In addition, the ensemble methods improve the accuracy of the system significantly, in which the best ensemble method is ensemble selection, which achieves 69.3% for subtask 1A and 63.6% for subtask 1B. This improves the baseline system by 8.9% and 14.4% for subtask 1A and 1B, respectively.

Index Terms— DCASE 2018, acoustic scene classification, convolution neural network, nearest neighbor filter.

1. INTRODUCTION

Acoustic scene classification (ASC) is defined as recognition of the environment based on the acoustic scene which is assumed to be a valid characterization of a location or situation. Furthermore, it is assumed to be distinguishable from other scenes based on its acoustic properties [1]. Sound events are introduced as important descriptors for an acoustic scene [2], however,the sound events are complex and can have a high degree of overlap. In real environments, sounds are unstructured and often unpredictable in its occurrence [3] causing more challenges for ASC compared to speech and music signal processing. However, the motivation for recent research on ASC is in designing a system that is able to capture and exploit the specific properties of a given audio scene. These algorithms are embedded in commercial smart devices with microphones to recognize acoustic contextual information.

Up to now, the basic framework of ASC includes feature extraction and classification that have been the crucial stages contributing to the effectiveness of an ASC algorithm. The most popular features applied in the ASC are representations of mel-frequency scales such as mel-frequency cepstral coefficients (MFCCs) and log-mel energies [4], [5]. According to [6], the main reason for their success is that they provide a reasonably good representation of the spectral properties of the signal. Furthermore, a reasonably high inter-class variability allows for class discrimination. Beside that, these features can be used as basis for higher level features. For example, Recurrent Quantification Analysis (RQA) and I-vectors are features obtained from MFCCs by applying recurrent quantification analysis [7] and joint factor analysis (JFA) [4]; Histogram of Gradient (HOG), Linear Binary Pattern (LBP) are well-known image processing techniques that were also used for feature extraction based on various types of spectrograms and MFCCs [8], [9], [10]. Moreover, in order to better cover the characteristics of environmental sounds, low level features such as zero-crossing, spectral centroid, bandwidth, energy have been with high level features such as Label Tree Embeding (LTE) [11], [12].

For classification, conventional classifiers such as Gaussian Markov Models (GMMs), Hidden Markov Models (HMMs), Support Vector Machines (SVMs) and Neural Networks (NNs) were applied in almost all submitted reports in DCASE 2013, where no algorithms involving Deep Neural Networks (DNNs) had been used [6]. In DCASE 2016, beside conventional classification methods, many participants applied DNNs such as Convolution Neural Networks (CNNs), Recurrent Neural Networks (RNNs) or combinations of DNNs and GMMs, and HMMs [13], [14] or combinations of CNNs and RNNs [15]. In DCASE 2017 and recent works, deep learning has been even more effective [16], [17], e.g. Generative Adversarial Networks (GANs) have been the most successful system for ASC in DCASE 2017. They have been combined with SVMs for classification [5].

This paper introduces an ASC system which is applied for task 1A and task 1B of the DCASE 2018 challenge [1]. In order to extract more information of the acoustic scene, we use 128 log-mel energies of the spectrogram and additionally apply nearest neighbor filtering (NNF)[18]. Both types of features are considered in the CNNs. All features are preprocessed by splitting the acoustic scene into chunks of 1s. Finally, ensemble methods are applied to combine several features and CNN settings to provide a vote for the 10s data chunks.

The remainder of this paper is organized as follows. Section 2 explains details of the proposed system. Section 3 discuss the

*Thanks to Vietnamese - Austrian Government Scholarship for funding.
†Thanks to Austrian Science Fund.

[1] `http://dcase.community/challenge2018`

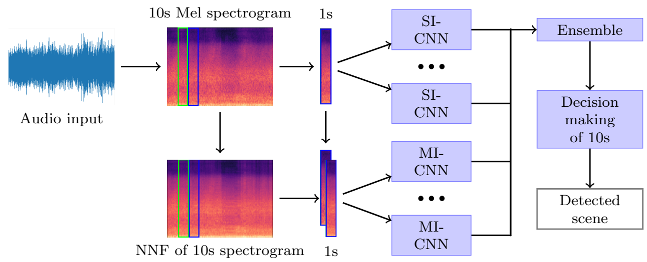

Figure 1: Illustration of the proposed algorithm.

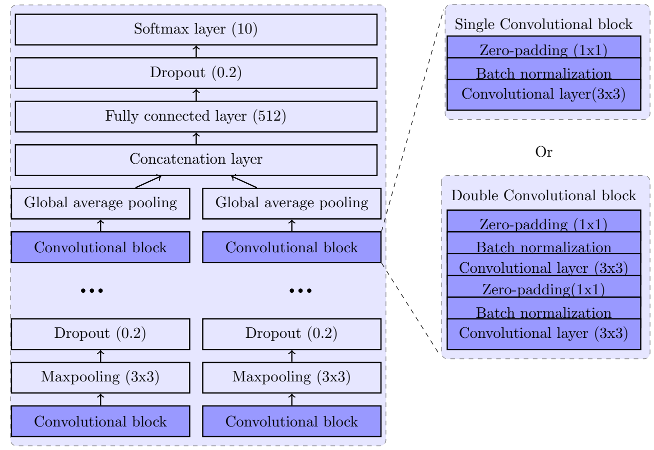

Figure 2: Illustration of the MI-CNN, single and double convolutional blocks.

experiments and results. Finally, conclusion is provided in Section 4.

2. PROPOSED SYSTEM

The proposed system is illustrated in Fig.1. The system is composed of 3 stages. First, the audio signal is converted to various time-frequency representations in 1s chunks. These features are then fed to the CNNs for training the models. Finally, probability outputs of 10 1s chunks of the CNN ensembles are used to produce the scene labels.

2.1. Audio Preprocessing

First, 128 bin mel-energies of the audio input are extracted. According to [16], it is important to keep a sufficient number of bins for representing the spectral characteristics while greatly reducing the feature dimensions. Window size for short-time Fourier transform is selected as 40ms and 20ms for hop size. We keep the sampling rating 48kHz for subtask 1A and 44.1 kHz for subtask 1B. In order to generate additional features for MI-CNNs, the mel-spectrogram is processed by a nearest neighbor filter [18]. Both the energies of the spectrogram and the filtered spectrogram are converted into logarithmic scale and normalized by subtracting the mean value and dividing by the standard deviation. The normalization step is determined feature-wise on the training set and parameters obtained are used to scale both training set and test set. The 10s audio files are processed in 1s audio chunks without overlap and fed to the CNN model as samples.

2.2. Nearest neighbor filter

Environmental sounds are often unstructured, neither predictable repetitions nor harmonic sounds [3] that are compounded by sound events and by overlapping of 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.

In our ASC system, we use nearest neighbor filters based on Repeating Pattern Extraction Technique (REPET) [18] for cases where repetitions happen intermittently or without a fixed period. The features are processed from the spectrogram as follows:

1. Compute a similarity matrix from the spectrogram using a similarity measure such as cosine, euclidean, L1, L2 or manhattan distance.
2. Identify the most similar frames in the spectrogram by using the similarity matrix as the reference.
3. Assign the median value of the identified frames for each frequency band to generate the filtered spectrogram.

Empirically, we observed that the euclidean distance is better than cosine distance and the number of nearest-neighbors for each sample is set to 5.[2]

2.3. Multi-input Convolution Neural Network

MI-CNNs have been used for ASC with different input features or structures of each branch of the CNN architecture. For example, in [20], authors used their CNN model as a “parallel” CNN architecture with different filter sizes and max-pooling sizes. In [15], they used a combination of Long Short Term Memories (LSTMs) and CNNs as feature extraction steps for each branch of their CNN model. In addition, according to [16], their CNN model used left-right (LR), *L+R* and *L-R* (MS), or harmonic-percussive source separation pairs as different input sources.

Our MI-CNN is inspired by these works. We feed 128 log-mel energies to one input branch of the CNN and their nearest neighbor filtered version to another one with the same CNN structure. Subsequently, we concatenate both branches before the fully-connected layer. Because the size of each sample is small i.e. 128 bins x 50 frames, 1x1 zero-padding is added to each convolution step in order to ensure that the whole data is processed. We proposed to use either a single convolutional block or a double convolutional blocks. A convolutional block consists of zero-padding, batch normalization and convolution layers, in which Rectifier Linear Units (ReLus) are used as activation function. The single/ double convolutional block is followed by a max-pooling layer and a dropout layer for the purpose of reducing dimensionality of the convolutional output and to ease the computation for upper layers as well as to reduce over-fitting in the training phase. Specifically, the last convolution blocks of the input branches are followed by global averaging pooling (GAP) instead of max-pooling and dropout. The structures of

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

the MI-CNN using single convolutional blocks and double convolutional blocks are shown in Fig. 2.

Beside that, we use CNN structures for single-input channel (SI-CNNs) using only an input branch. Empirically, we select the number of filters of convolutional layers for the CNNs including 2, 3 and 4 single or double convolutional blocks at 32 - 256, 32 - 64 - 256 and 32 - 64 - 128 - 256, respectively.[3]

2.4. Convolutional neural network Ensemble

Ensembles of CNNs combine the output probabilities of CNNs in order to improve performance [21]. The CNNs in the ensemble are trained individually and then their outputs are combined by voting, averaging, weighed averaging or model selection with and without replacement [22].

We compared performance of three ensemble methods named average ensemble (AE), weighted average ensemble (WE) and ensemble selection with replacement (ES). Basically, the similarity of these ensemble methods is that the output probabilities from all CNNs are averaged before making predictions. However, they are different in determining the contribution levels of each model to the ensemble using weights. Fig.3 shows the general architecture of the ensemble. Average ensemble is a simple ensemble where the output probabilities from all CNNs are equally weighted and averaged. The constraint of the weights is to be equal for all CNNs and sum to one. Weighted average ensemble and ensemble selection are more complex and introduced different weight values. Weighted average ensemble determines the optimal weights by minimization of the cross-entropy loss of ground-truth labels and estimated labels with constraints of the weights to sum to one. Ensemble selection with replacement [22] is an iterative method that allows models to be added to an ensemble multiple times such that the performance of the combination is maximized. The model weights are equivalent to the number of times the model has been selected divided by the total number of models in the ensemble.

We use the test data to determine the optimal weights for WE and ES. Sequential Least Squares Programming (SLSQP) is used for optimization of WE. For ES, we start with the best model among 12 candidate models in the ensemble before greedy step-wise selection of 200 iterations is performed. There are a significant difference between the weight values of WE and ES. These optimal weights are used for evaluation.

In addition, we try majority voting (MV) in which the output probabilities of every 1s chunk is binarized to “0” and “1” with the global threshold at 0.5. Majority voting determines the class which occurs most often among 10 1s chunks of an audio file. For average voting (AV) we use the *argmax* on the mean of the probabilities over 10 s. The experimental results shows that AV nearly always outperforms MV.

3. EXPERIMENTS

3.1. Data

The audio dataset for the ASC task of DCASE 2018 includes two different versions, TUT Urban Acoustic Scene 2018 and TUT Urban Acoustic Scene 2018 Mobile recorded in six European cities for 10 scenes. The former dataset is used for subtask 1A where the development and evaluation data are recorded by the same device.

[3]The CNNs are implemented on Keras `https://github.com/keras-team/keras`

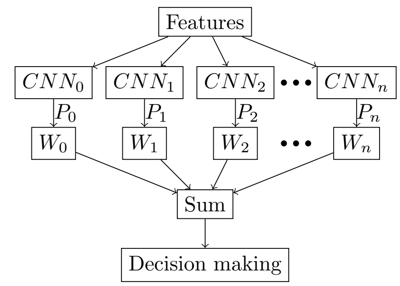

Figure 3: Architecture of CNN ensemble.

While the later one is used for subtask 1B in which the development set is comprised of subtask 1A dataset resampled and averaged into a single channel and a small amount of data is recorded by other devices. The original recordings were split into 10-second segments that are provided in individual files.

The subtask 1A dataset includes 8640 segments with 6122 segments for training and 2518 segments for testing. The subtask 1B training subset contains 6122 segments from device A, 540 segments from device B, and 540 segments from device C. The test subset contains 2518 segments from device A, 180 segments from device B, and 180 segments from device C.

3.2. Setup

The validation set accounts for approximately 30% of the original training data. We use a balancing mode for separation such that 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 nearest neighbor filtered version with 40 ms analysis frame and 50% hop size. The network training is carried out by optimizing the categorical cross-entropy and the Adam optimizer at learning rate of 0.001 is used. We use Glorot uniform data to initialize the network weights. The number of epochs and batch size was 500 and 16, 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.[4]

3.3. Performance on the test set

Table 1 present the accuracy of subtask 1A and subtask 1B for the different SI-CNNs (SI_) and MI-CNNs (MI_) using majority voting (_MV) and average voting (_AV) methods. The CNNs consists of various numbers of single convolutional blocks (_s) or double convolutional blocks (_db) as well as dropout layers (_D) and no dropout layers (_NoD). The performances of different ensemble methods of the 12 models are also presented. For determining the weights of ES and WE the labels of the test set are used.

According to results of Table 1, we can see that systems using the average voting method almost always performs better compared to majority voting. Results of average ensemble (AE_) and weighted

[4]Thanks to the DCASE organizers for providing the baseline system source code and the DCASE-UTIL toolbox `https://github.com/DCASE-REPO`

Table 1: Accuracy of proposed models and of ensemble methods using majority voting and average voting with and without dropout.

Algorithms	1A_MV	1A_AV	1B_MV	1B_AV
Baseline	59.7 ± 0.7	-	45.6 ± 3.6	-
SI_s_2cnn_NoD	61.1	62.3	54.2	56.1
SI_s_3cnn_NoD	64.3	65.0	56.9	57.5
SI_s_4cnn_NoD	63.9	64.7	53.1	54.4
SI_db_2cnn_NoD	63.6	64.4	57.5	58.9
SI_db_3cnn_NoD	63.0	64.1	59.2	60.6
SI_db_4cnn_NoD	64.3	65.3	51.4	53.6
MI_s_2cnn_NoD	61.0	62.1	51.1	52.2
MI_s_3cnn_NoD	64.5	64.4	54.2	55.3
MI_s_4cnn_NoD	62.7	63.4	54.2	54.7
MI_db_2cnn_NoD	66.3	66.8	53.6	55.6
MI_db_3cnn_NoD	63.6	64.0	57.5	56.4
MI_db_4cnn_NoD	63.1	63.2	52.8	52.5
AE_NoD	62.7	66.8	54.4	62.2
WE_NoD(*)	63.4	66.9	54.2	62.5
ES_NoD(*)	63.8	68.5	52.5	63.1
SI_s_2cnn_D	62.7	63.5	57.8	57.8
SI_s_3cnn_D	65.4	65.6	58.1	58.3
SI_s_4cnn_D	63.1	62.9	54.7	55.8
SI_db_2cnn_D	64.3	64.5	60.3	62.2
SI_db_3cnn_D	64.9	65.2	54.4	55.8
SI_db_4cnn_D	64.3	64.6	53.1	54.4
MI_s_2cnn_D	63.8	64.4	54.2	56.9
MI_s_3cnn_D	63.9	64.4	52.8	53.9
MI_s_4cnn_D	61.9	62.6	56.7	56.4
MI_db_2cnn_D	63.5	64.0	55.0	54.4
MI_db_3cnn_D	64.3	64.3	55.3	56.1
MI_db_4cnn_D	65.2	65.8	52.5	53.1
AE_D	63.5	67.4	53.9	61.4
WE_D(*)	65.3	68.3	54.2	61.7
ES_D(*)	65.5	**69.3**	56.7	**63.6**

average ensemble (WE_) are nearly the same and lower than of ensemble selection (ES_). Furthermore, dropout slightly improves the performances.

3.4. DCASE 2018 submission

The differrent submissions for subtask 1A and subtask 1B using average voting are:

- **1A_ES_D**: Ensemble selection of the 12 CNN models including dropout layer for task 1A.
- **1B_ES_D**: Ensemble selection of the 12 CNN models including dropout layer for task 1B.

Class-wise accuracy of both are represented in table 2.

4. CONCLUSION

In this paper, we proposed ensembles of 12 CNN structures in order to enhance the classification accuracy for subtask 1A and subtask 1B of DCASE 2018 challenge. We obtained performances of 69.3% for subtask 1A and 63.6% for subtask 1B on the test set by using ensemble selection combined with average voting. Additionally, we introduce nearest neighbor filtering for MI-CNN structures, which emphasizes the sound events in a scene. In particular, the nearest neighbor filters are applied on 128 log-mel energies of the spectrogram. The experimental results show that in subtask 1A the combination of the MI-CNN structures and pairs of log-mel features and their nearest neighbor filtering get higher performance than that of SI-CNN models using log-mel features. The highest accuracy corresponding to specific models are 66.8% and 65.6% respectively, while for subtask 1B the accuracy of the MI-CNN system is 56.9% compared to 62.2% for the SI-CNN system.

Table 2: Class-wise accuracy of submissions on the test set for subtask 1A and 1B.

Algorithms	1A_ES_D	1B_ES_D
Airport	75.8	58.3
Bus	73.1	80.6
Metro	57.9	41.7
Metro station	76.1	61.1
Park	83.9	91.7
Public square	58.3	55.6
Shopping mall	41.9	75.0
Street_pedestrian	57.5	50.0
Street_traffic	88.6	83.3
Tram	80.1	38.9
Average	**69.3**	**63.6**

5. 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.

6. REFERENCES

[1] A. Mesaros, T. Heittola, A. Diment, B. Elizalder, A. Shah, E. Vincent, B. Raj, T. Virtanen, " DCASE 2017 challenge setup: Tasks, datasets and baseline system," DCASE2017 Challenge, Tech. Rep., Nov. 2017.

[2] T. Heittola, A. Mesaros, A. Eronen, T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech and Music Proceeding*, 2013.

[3] S. Chu, S. Narayanan and C. Jay Kuo "Content analysis for acoustic environment classification in Mobile Robots," *Proc. AAAI Fall Symposium*, Oct. 2006.

[4] 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. Sep. 2016.

[5] S.Mun, S. Park, D. Han and H. Ko, "Generative Adversarial Network Based Acoustic Scene Training Set Augmentation and Selection Using SVM Hyper-Plane," DCASE2016 Challenge, Tech. Rep. Sep. 2017.

[6] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen and M. D. Plumbley, " Detection and Classification of Acoustic Scenes and Events: Outcome of the DCASE 2016 Challenge," *IEEE/ACM Trans. Audio, Speech, and Language Process.*, vol. 26, pp. 379-393, Nov. 2017.

[7] G. Roma, W. Nogueira, P. Herrera, and R. De-boronat, "Recurrence quantification analysis features for auditory scene classification," *Proc. IEEE AASP*, 2013, pp. 1 - 4.

[8] V. Bisot, S. Essid and G. Richard, "HOG and subband power distribution image features for acoustic scene classification," *Proc. EUSIPCO*, pp. 719-723, 2015.

[9] A. Rakotomamonjy and G. Gasso, "Histogram of Gradients of Time-Frequency Representations for Audio Scene Classification," *IEEE/ACM Trans. Audio, Speech, and Language Process.* , vol. 23, no. 1, pp. 142-153, 2015.

[10] S. Abidin, R. Togneri and F. Sohel, "Enhanced LBP texture features from time frequency representations for acoustic scene classification," *Proc.IEEE ICASSP* 2017, pp. 626-630.

[11] H. Phan, L. Hertel, M. Maass, P. Koch and A. Mertins, "CNN-LTE: a Class of 1-X Pooling Convolutional Neural Networks on Label Tree Embeddings for Audio Scene Recognition," DCASE2016 Challenge, Tech. Rep. Sep. 2016.

[12] A. Dang, T. H. Vu and J. .C. Wang, "Acoustic scene classification using convolutional neural networks and multi-scale multi-feature extraction," *Proc. ICCE*, pp. 1-4, 2018.

[13] G. Takahashi, T. Yamada, S. Makino and N. Ono, "Acoustic Scene Classification Using Deep Neural Network and Frame-Concatenated Acoustic Feature," DCASE2016 Challenge, Tech. Rep., Sep. 2016.

[14] G. Marques and T. Langlois, "TUT Acoustic Scene Classification Submission," DCASE2016 Challenge, Tech. Rep., Sep. 2016.

[15] S. Bae, I. Choi and N. Kim, "Acoustic Scene Classification Using Parallel Combination of LSTM and CNN," DCASE2016 Challenge, Tech. Rep. Sep. 2016.

[16] Y.Han and J. Park, " Convolutional Neural Networks with Binaural Representations and Background Subtraction for Acoustic Scene Classification," DCASE2017 Challenge, Tech. Rep. Sep. 2017.

[17] B. Lehner,H. Eghbal-zadeh, M. Dorfer, F Koreniowski, K. Koutini and G. Widmer, " Classifying Short Acoustic Scenes with I-Vectors and CNNs: Challenges and Optimisations for the 2017 DCASE ASC Task," DCASE2017 Challenge, Tech. Rep. Sep. 2017.

[18] Z. Rafii and B. Pardo,"Music/voice separation using the similarity matrix," *Proc. ISMIR* , pp. 583-588, 2012.

[19] M. Brian, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in python," in*Proc. the 14th Python in Science*, pp. 18-25. 2015.

[20] T. Lidy and A. Shindler, "CQT-Based Convolutional Neural Networks for Audio Scene Classification and Domestic Audio Tagging," DCASE2016 Challenge, Tech. Rep. Sep. 2016.

[21] X.Frazao and L. Alexandre, " Weighted Convolutional neural Network Ensemble," Bayro-Corrochano E., Hancock E. (eds) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2014. Lecture Notes in Computer Science, vol 8827. Springer, Cham.

[22] R. Caruana, A. Niculescu-Mizel, G. Crew and A. Ksikes, " Ensemble Selection from Libraries of Models," *Proc. ICML*, pp. 18-, Canada, 2004.