ACOUSTIC SCENE CLASSIFICATION WITH MULTIPLE INSTANCE LEARNING AND FUSION

Technical Report

Valentin Bilot, Ngoc Q. K. Duong, Alexey Ozerov

InterDigital R&D France
email: {valentin.bilot, quang-khanh-ngoc.duong, alexey.ozerov}@interdigital.com

ABSTRACT

Audio classification has been an emerging topic in the last few years, especially with the benchmark dataset and evaluation from DCASE. This paper present our deep learning models to address the acoustic scene classification (ASC) task of the DCASE 2019. The models exploit multiple instance learning (MIL) method as a way of guiding the network attention to different temporal segments of a recording. We then propose a simple late fusion of results obtained by the three investigated MIL-based models. Such fusion system uses multi-layer perceptron (MLP) to predict the final classes from the initial class probability predictions and obtains a better result on the development and the leaderboard dataset.

***Index Terms*—** DCASE 2019, acoustic scene classification, convolutional neural network (CNN), multiple instance learning (MIL), attention module.

1. INTRODUCTION

In this paper we will present the technical details of the systems submitted to Task 1, subtask A of the DCASE 2019 challenge[1]. This subtask focuses on acoustic scene classification of data from the same device as the available training data. We submitted four systems, all are based on the MIL framework which allows the trained DNN models to have different attention levels to different audio segments. Details of each system are described in Section 2.

2. MIL-BASED MODELS FOR ASC

The global workflow of our MIL-based systems is shown in Figure 1. To take into account multichannel setting, the systems first take mel-spectrogram of left (L), right (R) channel and their difference (R-L) as input. The original 10 second length signal is split into 10 segments of 1 second each, thus a *bag* in MIL framework here contains 10 *instances*. As a result, the input to CNN layers has size (10,3,100,50) where 100 stands for the number of mel frequency bin and 50 is the number of time frame. We investigated the use of three different CNN architectures for our three submitting systems as follows.

- Model 1: This model is mostly inspired from the baseline provided by DCASE challenge, but adapted for the different input size. This CNN is composed of 2 layers of convolution with 32 and 64 channels, and a kernel size of 7. The pooling layer after the first convolution has a kernel of size (5,5) and the second one a kernel of size (4,10).
- Model 2: This model is based on data driven activation function for the convolution layers [1]. There are 3 layers of convolution, with 64, 128 and 256 channels, with kernels of size (13,8), (6,5) and (5,4). Between each layer of convolution there is a convolution layer with kernel of size (3,3) whose results goes in a sigmoid function and then multiplies with the result of the main convolution layer, this multiplication replaces the ReLU activation function. The goal of this operation is to give the network the ability to focus on certain patterns more intensively.
- Model 3: This model is based on dilated convolution network [2]. It also have gates like the Model 2 and consists of 3 layers of convolution with kernels of size (5,5), dilation of (3,5) for the first two layers and (1,1) for the last layer, and the number of channels are 16, 32 and 64, respectively. The number of channels is limited here because the purpose of this model is to limit complexity in the same time of having a wide feature detection area.

So far, we have extracted CNN features for each 1 second temporal instance. We then use the two-stream architecture proposed by Bilen *et al.* [3] for weighting each of them with respect to the classes. This kind of attention module consists of parallel classification and localization streams. The former classifies each instance by passing the CNN feature through a linear fully connected layer with weights W^a_{cls}. On the other hand, the localization layer passes the same feature through another fully-connected layer with weights W^a_{loc}. This is followed by a softmax operation over the resulting matrix which allows the localization layer to choose the most relevant proposals for each class. Subsequently, the classification stream output is weighted by the attention weights through element-wise multiplication. This MIL-based architecture has been discussed also *e.g.,* in [4, 5] for audio classification. However, while in [4] and [5] the class scores over the whole audio file are obtained by summing the resulting weighted scores, we concatenate all the score vectors as input to a higher MLP layer and we found that this strategy offers a slightly higher classification accuracy.

2.1. Audio preprocessing

We use mel-spectrograms as input feature to the DNN as it has been shown to be state-of-the-art features for audio classification tasks and used in the baseline system. For the short-term Fourier transform (STFT), we used a 40 ms length Hann window with 50% overlap. The resulting mel-spectrogram for each 1 second audio

[1] http://dcase.community/challenge2019/task-acoustic-scene-classification

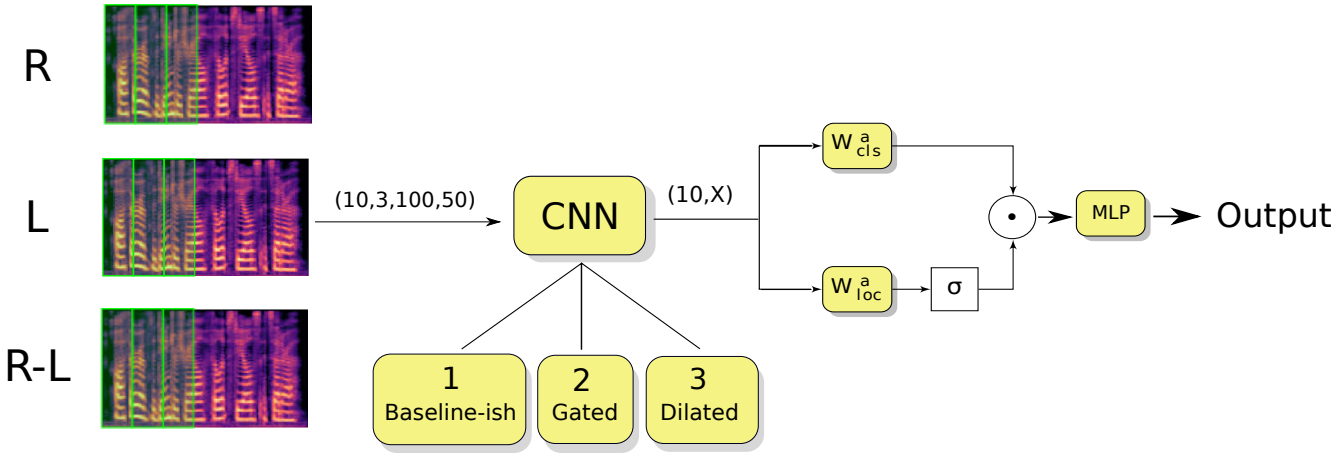


Figure 1: Overall workflow of the three MIL-based systems. Model 1 uses the baseline's CNN architecture, Model 2 uses gated CNN layers while Model 3 uses dilated CNN layers.

segment has size of (100, 50) where 100 stands for the number of mel frequency bins and 50 the number of time frame. Dorfer team in DCASE2018 demonstrated that using left, right and difference channels is an easy and efficient way of taking advantage of stereo inputs [6], so we adopted the same strategy. As a resut, the final input to CNN layers has size of (10,3,100,50).

2.2. Data Augmentation

We tried several ways for data augmentation such as noise addition, pitch shifting and time stretching, but we did not observe the improvement in the overall classification result. In the end only mixup strategy [7] brings little benefit to our systems. Mixup creates a new training pair (sample, label) (X, y) by mixing two training pairs (X_1, y_1) and (X_2, y_2) as:

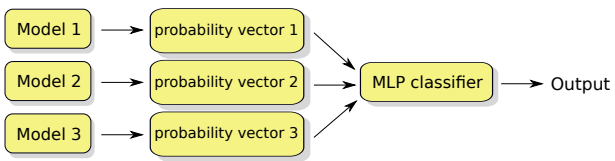
$$X = \lambda X_1 + (1 - \lambda)X_2 \quad (1)$$

$$y = \lambda y_1 + (1 - \lambda)y_2 \quad (2)$$

In our implementation, we choose 60% mixed up samples in each batch and $\lambda = 0.8$

3. MODEL FUSION

As each studied model can perform differently for each class, we investigate in addition the late fusion of all the three models. For this purpose, given an input audio file, the class probability vectors predicted by each model are concatenated and passed through a MLP layer, followed by a softmax to predict the final class as shown in Figure 3. This ensemble learning is shown to provide better classification result on the development dataset and the leaderboard dataset in our test.



Late model fusion strategy by MLP.

4. EXPERIMENT

4.1. Dataset

The provided development dataset is the TAU Urban Acoustic Scenes 2019 dataset [8], which consists of recordings from the same device for 10 known acoustic scene classes. It consists of about 40 hour recordings, balanced between classes in a total of 10 large European cities. For each scene class, recordings were done in different locations; for each recording location there are 5-6 minutes of audio. The original recordings were split into 10 second segments that are provided in individual files with an associated label.

This development dataset is then split into the training subset contains recordings from only 9 of the cities, to test the generalization properties of the systems. The training/test subsets are created based on the recording location such that the training subset contains approximately 70% of recording locations from each city. The test subset contains recordings from the rest of the locations, and few locations from the tenth city. Overall, the dataset contains 14400 segments (144 per city per acoustic scene class).

The leaderboard dataset consists of a small subset of the official evaluation dataset, with similar properties (distribution). The material amount in the leaderboard dataset is considerably lower than the official evaluation material in the DCASE challenge, 1200 samples for the leaderboard dataset and 7200 samples for the evaluation dataset.

4.2. Training

All trainings are done on GPU, with a batch size of 64, with the binary cross entropy for the loss function, and with RMSProp for the optimizer. Model 1 is trained with a learning rate of 10^{-3} and Models 2 and Model 3 are trained with a learning rate of 10^{-4}.

The three models are firstly trained on the train set provided by the challenge (fold 1) and early stopped with the corresponding validation set. Then the MLP of the fusion model is trained on the results of those first three models on the validation set and early stopped on the test set.

In order to increase the number of data seen by the first three models for their independent submission, we also trained the first

	Train	Validation	Test	Leaderboard
Model 1	84.8	65.1	67.3	69.7
Model 2	96.8	70.9	70.6	71.0
Model 3	73.4	64.2	64.1	66.7
Fusion	x	65.2	72.3	72.7
Baseline	x	x	62.5	63.0

Table 1: Average accuracy of the four submitted systems on the training, validation, and test set split from the development dataset.

Classes	Model 1	Model 2	Model 3	Fusion	Baseline
Airport	59.8	71.9	55.1	72.2	48.4
Bus	75.2	74.2	62.6	76.1	62.3
Metro	64.9	62.3	51.5	65.8	65.1
Metro station	58.4	62.1	39.5	61.6	54.5
Park	85.0	77.8	80.1	81.6	83.1
Public square	61.2	54.7	61.4	54.5	40.7
Shopping mall	64.4	65.7	74.0	69.4	59.4
Street pedestrian	60.8	80.4	65.5	79.7	60.9
Street traffic	86.3	89.0	86.3	90.5	86.7
Tram	57.7	67.6	64.2	69.9	64.0
Overall	67.3	70.6	64.1	72.3	62.5
Leaderboard	69.7	71.0	66.7	72.7	63.0

Table 2: Classwise and overall accuracy on test set of the four submitted systems, when compared to the Baseline from the DCASE challenge.

three models on the evaluation set and early stopped them on the test set.

4.3. Results

The average accuracy for all classes obtained by our four submitted systems on the training, validation, and test set split from the development dataset (fold 1), and the leaderboard is shown in Table 1. As can be seen, the results for training are significantly higher than that for the evaluation and test set. This reveals a possible overfiting issue even thought we tried different forms of regularization such as dropout and reducing the model size. Model 2 obtained the higher accuracy than the Model 1 and Model 3 while the fusion of all three models (named Fusion) reached the best result (*i.e.* 72.3% accuracy) on the test set. It is also worth noting that all our four systems outperform the Baseline from DCASE challenge on such development dataset and the leaderboard dataset.

For more details on how each system performs in each class, the classwise and overall accuracy of the four systems on the test set, when compared to the Baseline from the DCASE challenge is provided in Table 2. Table 3 shows the confusion matrix of the fusion system on the split test set. As can be seen, some pairs of classes are quite confusing for the prediction systems such as bus/tram, air/shop, street/square.

5. CONCLUSION

This report presents our four machine learning models for acoustic scene classification submitted to the Task 1a of the DCASE 2019 challenge. The first three models are based on MIL framework with the use of CNN layers for feature learning. The fourth model exploits the late fusion of results obtained by the first three model and

	air.	shop.	metro sta.	street ped.	p. square.	str. trf.	tram	bus	metro	park
air.	**310**	78	26	8	8	0	1	0	5	0
shop.	78	**306**	46	28	3	0	1	0	2	0
metro sta.	19	14	**268**	10	4	1	5	4	49	0
street ped.	13	41	34	**342**	123	6	1	0	0	11
p. square.	0	2	7	36	**211**	24	1	0	2	49
str. trf.	1	0	3	4	23	**364**	1	0	0	11
tram	0	0	9	0	0	0	**305**	74	63	0
bus	0	0	7	0	0	0	51	**316**	26	0
metro	0	0	34	0	1	0	68	20	**285**	0
park	0	0	1	1	14	7	2	1	1	**315**

Table 3: Confusion matrix of the fusion model on the test set.

shown to provide better classification performance on the investigated dataset. We are looking forward to seeing the final results obtained on the official test set of the DCASE challenge.

6. REFERENCES

[1] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language Modeling with Gated Convolutional Networks," *arXiv:1612.08083 [cs]*, Dec. 2016, arXiv: 1612.08083. [Online]. Available: http://arxiv.org/abs/1612.08083

[2] Z. Ren, Q. Kong, J. Han, M. D. Plumbley, and B. W. Schuller, "Attention-based Atrous Convolutional Neural Networks: Visualisation and Understanding Perspectives of Acoustic Scenes," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 56–60.

[3] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *CVPR*, 2016, pp. 2846–2854.

[4] C. Yu, K. S. Barsim, Q. Kong, and B. Yang, "Multi-level attention model for weakly supervised audio classification," *arXiv preprint arXiv:1803.02353*, 2018.

[5] S. Parekh, S. Essid, A. Ozerov, N. Q. K. Duong, P. Perez, and G. Richard, "Weakly supervised representation learning for unsynchronized audio-visual events," in *CVPR Workshop*, 2018.

[6] M. Dorfer, B. Lehner, H. Eghbal-zadeh, H. Christop, P. Fabian, and W. Gerhard, "Acoustic scene classification with fully convolutional neural networks and I-vectors," DCASE2018 Challenge, Tech. Rep., September 2018.

[7] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," Oct. 2017. [Online]. Available: https://arxiv.org/abs/1710.09412v2

[8] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13. [Online]. Available: https://arxiv.org/abs/1807.09840