ACOUSTIC SCENE CLASSIFICATION BY THE SNAPSHOT ENSEMBLE OF CNNS WITH XGBOOST

Technical Report

Reyhaneh Abbasi

Austrian Academy of Science
Acoustic Research Institute
Wohllebengasse 12-14, A-1040, Vienna, Austria
reyhaneh.abbasi@oeaw.ac.at

Peter Balazs[†]

Austrian Academy of Science
Acoustic Research Institute
Wohllebengasse 12-14, A-1040, Vienna, Austria
peter.balazs@oeaw.ac.at

ABSTRACT

This report is to explain our system for the DCASE challenge 2020 task 1A. The aim is to implement acoustic scene classification of audio recordings into 10 predefined classes including airport, shopping mall, metro station, street pedestrian, public square, street traffic, tram, bus, metro, and park. There are main challenges to accomplish; 1- recordings are provided by six devices with different quality and 2- some of classes are very similar in terms of acoustic information. To bias correct all devices against the reference (here the device A), we have used XGBOOST algorithm fed by standardized Mel spectrogram. Our classifier consists of a CNN with mix-up augmentation and snapshot ensemble. The proposed model (baseline model) has yielded the accuracy of 62.1% (54.1%) and cross-entropy loss of 1.06 (1.36).

***Index Terms*—** Acoustic scene classification, XGBOOST, CNNs, snapshot ensemble, domain adaptation

1. INTRODUCTION

Task 1A of the DCASE 2020 challenge [1] is concerned with acoustic scene classification targeting generalization properties of systems across a number of mismatched recording devices. The study dataset is called “TAU Urban Acoustic Scenes 2020 Mobile”. For training phase, recordings are acquired by 6 different recording devices at ten different acoustic scenes within ten European cities. This dataset contains quite a few samples (10215 audios) from a high-quality device (referred to as A). Whereas each of other devices monitored only 750 samples (referred to as B, C, S1, S2, S3),

2. EXPERIMENTAL SETUP

2.1. Data Preparation

All audio files are resampled to the sampling rate of 44.1 kHz. We extracted the input features using a Short Time Fourier Transform (STFT) with a window length of 0.04 sec and hop length of 0.02 sec. Then, the spectrogram size of each audio sample was reduced from 1025*500 to 256*500, using a Mel-scaled filter bank of 256.

The resulting mel-spectrograms (hereafter named as Xmel) were row-wise standardized. Figure 1 shows three example Xmels produced for A, S2, and S3 when recording at tram. First and second rows are original and standardized Xmel, respectively. The latter could remove the background noise and make the Xmel of different devices more similar to each other.

2.2. Adjusting recording devices

In order to account for different frequency responses of the devices B, C, S1, S2, and S3 (hereafter called as target devices) compared to the device A (hereafter called as source device). we have trained XGBOOST regression system, known as an ensemble of regression trees [2], using 3750 aligned audio recordings in development dataset. Squared loss between A and predicted frequency responses is taken as the loss function. It is worth to mention that one XGBOOST model has been developed for each target device (5 models in total) using flattened Xmels. The final prediction is the ensemble (average) of these 5 models’ outputs. For example, the output of this adjusting model applied on the sample audio recording is shown in the bottom row of Figure 1. Interestingly, this model did not cause significant changes in frequency responses of the source device.

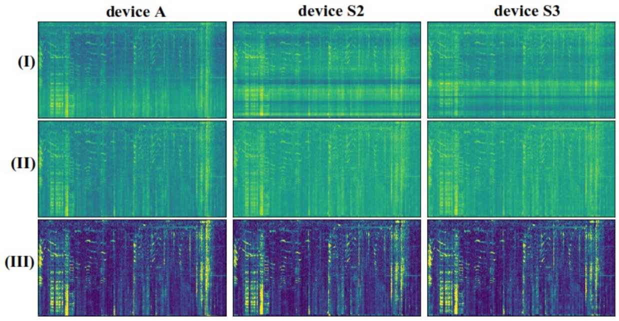

Figure 1: Mel-spectrogram of an audio sample from three devices A, S2, and S3 while recording at tram. Rows I, II,

[††] funded by START-project FLAME

and III are the original-, standardized-, and the XGBOOST-predicted mel-spectrograms.

2.3. Classifier

For the classification of (adjusted) audio scenes, we have applied convolutional neural networks (CNNs) [3]. The best CNNs architecture is determined based on performance (accuracy) of CNNs on validation dataset. The selected CNNs is trained over the whole development dataset. Table 1 summarized this architecture in which convolutional layers used BatchNormalization and ReLU activation [4]. Dropout with the rate of 0.3 was applied after each ReLU activation function layer.

Table 1: Architecture of the classifier

layer	**outputs**	**kernel**	**stride**
Conv2D + BN + ReLU	32	7	1
MaxPooling2D	32	5	-
Conv2D + BN + ReLU	64	7	1
MaxPooling2D	64	(4, 100)	-
Dense + ReLU	100	-	-
Dense + Softmax	10	-	-

CNNs is optimized using stochastic gradient descent [5] with momentum of 0.9 and categorical cross entropy as the loss function. Batch size and number of epochs, set to 64 and 300 respectively, were chosen by trial and error. We have used mix-up augmentation [6] with alpha of 0.2 to augment training dataset.

To use the advantages of Ensembling multiple neural networks, while avoiding additional training cost, we have used Snapshot Ensembling [7]. This method converges to several local minima along its optimization path and the model parameters are saved. We have saved one model after every 60 epochs. Therefore, there are 5 models saved in total. The schedule of used learning rate is shown in Figure. 2. The start learning rate is set to 0.01 which declines to 6.8e-6 based on a cosine function through 60 epochs. Updating learning rate to 0.01 is to escape the current local minimum of loss function.

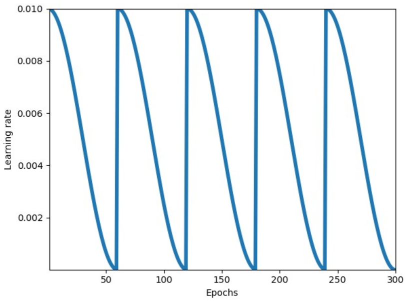

Figure 2: Schedule scheme used for learning rate

After testing different combinations, the final prediction is chosen to be the average of all 5 CNNs models.

3. RESULTS

The results presented here are the evaluation of the developed model on the test dataset. Test dataset consists of 330 samples from each target and source devices plus those from three new devices S4, S5, and S6. Table 2 shows the performance of the developed CNN classifier (with and without XGBOOST adjusting model) compared to the baseline model. The baseline model is provided by the challenge organizer [1].

Table 2: Accuracy and loss error calculated for the test dataset

Submitted system	**Model**	**Accuracy**	**Loss**
-	baseline	54.1%	1.365
Abbasi_ARI_task1a_1	CNNs without XGBOOST	61.60%	1.07
Abbasi_ARI_task1a_2	CNNs with XGBOOST	62.01%	1.06

Based on the table above, CNNs fed by Xmel which are adjusted with XGBOOST has yielded the best performance than two other studied model. This model not only has the best overall performance, but it outperformed two other models in all devices (not shown here).

4. CONCLUSION

In this technical report, we detailed our approaches to develop a classifier for the Task 1A of the DCASE-2020 challenge. We showed that snapshot ensemble of CNNs fed by XGBOOST regularized mel-spectrogram could yield the best performance compared to CNNs without XGBOOST adjusting model and baseline system.

5. REFERENCES

[1] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 Challenge: generalization across devices and low complexity solutions," *arXiv preprint arXiv:.14623,* 2020.

[2] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785-794.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[4] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Available: arXiv preprint arXiv:.03167,* 2015.

[5] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *Proceedings of COMPSTAT,* pp. 177-186, 2010.

[6] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *International Conference on Learning Representations (ICLR),* 2018.

[7] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get m for free," *International Conference on Learning Representations (ICLR)* 2017.