ACOUSTIC SCENE CLASSIFICATION USING MEL-SPECTRUM AND CQT BASED NEURAL NETWORK ENSEMBLE

Technical Report

Hong Lu

Intel Labs
Intel Corp
2200 Mission College Blvd.
Santa Clara, CA 95054, USA,
hong.lu@intel.com

ABSTRACT

In our submission to the DCASE 2020 Task1a, we have explored the use of ResNeXt-50 architecture with Log-Mel-spectrum and Constant-Q transform(CQT) based frontend. In order to improve performance, we use transfer learning technique. The neural networks were pre-trained with AudioSet data, and then fine-tuned over the DCASE task1a dataset.

With DCASE 2020 task1a default train/validation split, we got about 70% average accuracy across all the 10 classes. To further improve the performance, we applied a leave-one-city out cross validation(CV) method to train 10 more models, with one city's data as holdout set for each of the CV fold. These models were combined together with different ensemble strategies to produce 4 final submission entries.

***Index Terms*—** DCASE, Acoustic Scene Classification, Neural Networks,

1. INTRODUCTION

In the DCASE 2020 task1a[1], the dataset provided by the organizers is TAU Urban Acoustic Scenes 2020 Mobile. This dataset contains 10 acoustic scenes recordings from 12 different cities. It consists of recording from real devices and synthetic data created based on the original recordings.

The development set contains data from 10 cities, and the other 2 cities are presented only in the evaluation set. The challenge provide a default train/evaluation split for benchmark purpose.

Our proposed machine learning pipeline is a deep neural network, ResNeXt-50[2], with Log-Mel-spectrum and Constant-Q transform(CQT)[3] front end. An end-to-end audio processing pipeline is built with PyTorch[4] 1.5. For the frond end, we used nn-audio[5] library to compute Log-Mel-spectrum and CQT on GPU. Such that the pipeline takes raw wave form as input, and all the feature computation and classification are all handled and accelerated by GPU.

In the following sections, we discuss the design, implementation, and experimentation for our DCASE2020 challenge Task1a submission.

2. DATA PROCESSING

The DCASE2020 Task1a dataset contains 10-second audio clips in 44.1kHz 24-bit format. All the audio files come with city, location, and scene class labels as part of their file names.

Since we use Google Audioset[6] to pre-train our model for transfer learning. All audio files from both the DCASE2020 Task1a dataset and the audio set are resampled to the same format, i.e. 32kHz,16-bit wave format.

During the downsampling operation, the file names stay the same, so all the labels are retained for the DCASE2020 Task1a dataset dataset.

We employed a few standard audio data augmentation schemes, including, time stretching, adding noise, random cropping, mix-up[7], and spec-augmentation[8]. All the data augmentation operations are done on-the-fly within the pipeline itself.

3. CONVOLUTIONAL NEURAL NETWORK FOR ACOUSTIC SCENE CLASSIFICATION

Both Log Mel Spectrum and CQT have been widely used for audio scene classification. Actually, they have been used in previous DCASE challenge as well. We use nn-audio[5] library to compute Log-Mel-Spectrum and CQT. Then stack them together as the input to the neural network.

We chose the ResNeXt-50 as our neural network architecture. Our implementation is directly adapted from the implementation of torchvision library, with customized input and output layers. In particular, the input channel is reduce from 3 to 2, i.e. one channel for the Log-Mel-Spectrum and one channel for CQT. A batch norm layer is added right after the input layer. Finally, the FC layer are also expended and tailored for the 10-class output for the challenge task.

Relu is used through out the network as the activation function. Dropout and weight decay are used for regularization purpose, while batch norm, mix-up, and spec-augmentation also provide additional regularization effects. During training, cross entropy is used as the loss function. Standard Adam optimizer is used together with Cosine Annealing learning rate scheduler.

Fold #	Hold-out City	Accuracy
0	Barcelona	0.662
1	Helsinki	0.669
2	Lisbon	0.663
3	London	0.661
4	Lyon	0.702
5	Milan	0.710
6	Paris	0.718
7	Prague	0.719
8	Stockholm	0.677
9	Vienna	0.689

Table 1: Leave-One-City-Out Cross Validation Result

4. TRAINING STRATEGY

The pipeline is first pre-trained with 32kHz,16-bit AudioSet data, before it is fine-tuned over the DCASE task1a dataset. Durning the fine-tuning process, no layer is frozen and the gradients is let to back propagate through the whole network.

Using the pipeline described above, we trained 12 models on the DCASE 2020 task1a data set using two strategies.

- Two models are generated using the train/evaluation split provided with the development set. i.e. the audio files listed in fold1_train.csv are used for model training; the audio files listed in fold1_evaluate.csv are used for benchmark. During this phase, we conducted several rounds of experiments to search for the best hyper-parameters. After all the hyper-parameters are finalized, we trained 2 models. The two models generated 69.20% and 70.28% macro-average accuracy (average of the class-wise accuracies) score respectively on the default evaluation set.
- Ten models are trained with a leave-one-city-out cross-validation method. Instead of using the provided default train/evaluation split, the split is done on a per city basis, i.e. in each of the CV fold, nine cities in the development set are used for training, and one city is held out as the validation set. The performance on each of the validation fold is summarized on Table 1 .

5. ENSEMBLE STRATEGY

Ensemble has showed to be a powerful technique to reduce overfitting and generate overall better result. As we have the 12 models trained and benchmarked. We generated 4 submission entries using the following strategies.

- the best single model trained using default train/evaluation split provided by the challenge.
- the two models trained using default train/evaluation split provided by the challenge.
- the ten models trained with leave-one-city-out cross-validation method.
- all twelve models

When we combining multiple models together, we just average their inferred probability (output of their softmax layer) as the final output.

As this year the challenge does not provide any leaderboard or public benchmark test set, the performance of these strategies is unknown until the final result is released.

6. CONCLUSION

Using our deep learning pipeline, we were able to achieve performance above the baseline with a single model. The macro-average accuracy of our best single model we trained is 70.28% on the default train/evaluation split provided by the challenge.

Then multiple models are trained with different train/validation setups and ensemble them to generate 4 entries for DCASE challenge 2020 task1a. Using the ensemble technique, hopefully we could better handle overfitting and obtain models that could generalize well to unseen data from holdout cities in the test set.

7. REFERENCES

[1] http://dcase.community/challenge2020/.

[2] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” *CoRR*, vol. abs/1611.05431, 2016. [Online]. Available: http://arxiv.org/abs/1611.05431

[3] T. Lidy and A. Schindler, “Cqt-based convolutional neural networks for audio scene classification.”

[4] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[5] K. W. Cheuk, H. H. Anderson, K. Agres, and D. Herremans, “nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolution neural networks,” *ArXiv*, vol. abs/1912.12055, 2019.

[6] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[7] H. Zhang, M. Cisse, Y. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization. iclr 2018,” *arXiv preprint arXiv:1710.09412*, 2017.

[8] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” *arXiv preprint arXiv:1904.08779*, 2019.