CONVOLUTIONAL NEURAL NETWORKS WITH TRANSFER LEARNING FOR URBAN SOUND TAGGING

Bongjun Kim

Dept. of Computer Science
Northwestern University
Evanston, IL, USA
bongjun@u.northwestern.edu

ABSTRACT

This technical report describes sound classification models from our submissions for *DCASE challenge 2019-task5*. The task is to build a system to perform audio tagging on urban sound. The dataset has 23 fine-grained tags and 8 coarse-grained tags. In this report, we only present a model for coarse-grained tagging. Our model is a Convolutional Neural network (CNN)-based model which consists of 6 convolutional layers and 3 fully-connected layers. We apply transfer learning to the model training by utilizing VGGish model that has been pre-trained on a large scale of a dataset. We also apply an ensemble technique to boost the performance of a single model. We compare the performance of our models and the baseline approach on the provided validation dataset. The results show that our models outperform the baseline system.

Index Terms— DCASE, Urban sound tagging, Audio event classification, Deep learning, Convolutional neural networks

1. INTRODUCTION

This technical report describes our approach to the *DCASE challenge* Task5 (Urban Sound Tagging) [1]. The task of this challenge is to build a system to predict whether each of 23 (fine-grained) or 8 (coarse-grained) sources of noise pollution is present in a 10-second recording. This is a multi-label and multi-class classification problem. Our system in this technical report only predicts the 8 coarse-grained labels: engine, machinery-impact, non-machinery-impact, powered-saw, alert-signal, music, human-voice, and dog. As a development dataset, 2,351 audio files for training and 443 recordings for validation are provided with their labels. Since the 2,351 is a quite small number for model training, we perform transfer learning by using a part of an existing pre-trained CNN model. We also use an ensemble technique to boost the performance of the system.

2. DATASET

The challenge dataset is a small subset of urban noise recordings collected using the SONYC [1] acoustic sensor network for urban noise pollution monitoring. All recordings in the dataset are 10 seconds. Each recording contains sound events labeled as one of the pre-defined urban noise sources. The dataset provides two levels of hierarchical labels, 8 *coarse-grained* and 23 *fine-grained* labels. A 10-second recording can contain multiple sources and a single source can occur multiple times in a recording. For example, a recording contains a series of dog barking sound events as well as car engine sound. Therefore, the task in this challenge is a multi-label and multi-class classification. The dataset can be also thought as weakly-labeled dataset because it does not provide temporal information (i.e., onset and offset) of a sound event within a 10-second recording.

The development dataset for the challenge consists of 2,351 audio files for training and 443 recordings for validation. The evaluation dataset contains 274 recordings without their labels. While data augmentation and using external datasets are allowed in this task, none of them was used in our submission.

[1] `http://dcase.community/challenge2019/task-urban-sound-tagging`

Table 1: Model architecture for 8 coarse-grained classification. MP indicates 2D-Max Pooling (kernal size: 2×2, stride: 2). **MP* on Layer-6 is Max Pooling operation over time-axis

Layers	Components	Output shape
Input	Mel-spectrogram	998×64
Layer-1	Conv (3×3, 64) → Relu → MP	499×32, 64
Layer-2	Conv (3×3, 128) → Relu → MP	249×16, 128
Layer-3	Conv (3×3, 256) → Relu	249×16, 256
Layer-4	Conv (3×3, 256) → Relu → MP	124×8, 256
Layer-5	Conv (3×3, 512) → Relu	124×8, 512
Layer-6	Conv (3×3, 512) → Relu → **MP*	1×8, 512
Layer-7	FC (2048) → Relu	2048
Layer-8	FC (128) → Relu	128
Layer-9	FC (8) → Sigmoid	8

3. MODEL ARCHITECTURE

Table 1 shows the architecture of our model. It consists of 6 convolutional layers (Conv) and 3 fully-connected layers (FC). It takes a mel-spectrogram of 10-second audio as an input representation and its output is class probabilities of 8 noise classes. In the table, filter sizes and the number of channels of convolutional layers are represented as *Conv (width × height, the number of channels)*. The layer-1 to layer-6 were taken from VGGish model [2] which has been trained on 8M-YouTube dataset. The pre-trained model is

publicly available [2].

Our model takes an entire recording and makes clip-level predictions on the recording. The labels in the dataset are also clip-level *weak* labels. To compute clip-level loss and make clip-level predictions, frame-level feature maps should be pooled on time-axis. To do so, we applied a max-pooling operation to the output from layer-6 (before fully-connected layers). We have tried other pooling operations such as mean, softmax [3], and attention [4], but the max pooling showed the best performance in our models.

4. TRAINING PROCEDURE

Audio files for training are resampled to 16kHz mono. Each sample is represented by a log-scale Mel-spectrogram with 64 Mel bins, a window size of 25 ms and hop size of 10 ms. Given a 10-second audio file, the size of the input representation is 998×64.

We applied transfer learning when training our models. Layer-1 to layer-6 are initialized with parameters from convolutional layers of the pre-trained VGGish model. The three fully-connected layers are randomly initialized. During training, the first four convolutional layers (Layer-1 to 4) are fixed (not updated) and rest of layers are fine-tuned on the training set. The binary cross-entropy loss and Adam Optimizer with learning rates of 0.001 or 0.0001 were used. A model was trained for 50 epochs and the model that shows the lowest validation loss was chosen. We submitted three models to the challenge. The first and the second model (Model1 and Model2) are trained with learning rates of 0.001 and 0.0001 respectively. The third model (Model3) is an ensemble model which make predictions by averaging the outputs from Model1 and Model2.

5. EVALUATION

We evaluate our models on the provided validation dataset containing 443 recordings. As the classification metric, the challenge uses the Area Under the Precision-Recall Curve (AUPRC). To compute the curve, we first apply a fixed threshold τ to the confidence (model output) of every label in every recording to generate a one-hot encoding of predicted labels. Then, we count the total number of true positives (TP), false positives (FP), and false negatives (FN) between model prediction and ground-truth label over the entire dataset. More details of performance metrics can be found in the challenge website [3]. The source codes for the evaluation were also provided by the challenge organizers.[4].

We evaluated our three different models and the baseline system on the validation dataset. The baseline system is a logistic regression model which takes VGGish embeddings as its input representation. Table 2 shows class-wise AUPRC and total micro-averaged AUPRC for each model. It shows that our models outperform the baseline system for all coarse-grained categories. Among our models, the ensemble model (model3) shows the highest micro AUPRC.

6. CONCLUSION

We present a simple CNN architecture for urban sound tagging as a part of DCASE challenge 2019-task5. To train models effectively on the small number of training examples, we applied transfer learning using VGGish pre-trained model. While data augmentation and using external datasets are allowed in this challenge, none of them was used in our submission. As future works, after the labels for evaluation set is released, we will apply simple data augmentation techniques to see how much they help. Codes for training and testing models in this report are available. [5]

Table 2: AUPRC for the baseline system and our models. All three models outperforms the baseline model for all the coarse categories. Model3 is the ensemble model of model1 and model2.

Tag names	Baseline	Model1	Model2	Model3
engine	0.855	**0.856**	0.846	0.855
machinery-impact	0.360	0.570	**0.590**	0.586
non-machinery-impact	0.361	**0.574**	0.522	0.549
powered-saw	0.678	0.786	**0.815**	0.807
alert-signal	0.813	0.895	**0.910**	0.906
music	0.299	0.437	0.470	**0.485**
human-voice	0.945	0.975	0.972	**0.975**
dog	0.029	0.250	**0.292**	0.275
Total (Micro AUPRC)	0.762	0.837	0.840	**0.843**

7. REFERENCES

[1] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "Sonyc: A system for monitoring, analyzing, and mitigating urban noise pollution," *Communications of the ACM*, vol. 62, no. 2, pp. 68–77, Feb 2019.

[2] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "Cnn architectures for large-scale audio classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 131–135.

[3] J. Salamon, B. McFee, P. Li, and J. P. Bello, "Dcase 2017 submission: Multiple instance learning for sound event detection," Tech. Rep., 2017.

[4] Q. Kong, Y. Xu, W. Wang, and M. Plumbley, "Audio set classification with attention model: a probabilistic perspective," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[2] https://github.com/tensorflow/models/tree/master/research/audioset

[3] http://dcase.community/challenge2019/task-urban-sound-tagging

[4] https://github.com/sonyc-project/urban-sound-tagging-baseline

[5] https://github.com/bongjun/dcase2019-task5