AUTOMATIC AUDIO TAGGING WITH 1D AND 2D CONVOLUTIONAL NEURAL NETWORKS

Technical Report

Siyuan Shan

China
shansiliu@outlook.com

Yi Ren

China
yi.ren@dbsonics.net

ABSTRACT

In this work, we ensemble four different models for the audio tagging tasks. The first two models are 2D convolutional neural networks (CNNs) that respectively take Mel spectrogram and MFCC spectrogram as input features, while other last two models are two 1D CNNs architectures that take raw waveform as inputs. Data augmentation techniques, including time stretch, pitch shift, reverb and dynamic range compression, are also employed for better generalization. Transfer learning from two external datasets is also adopted. These components together contribute to our final recognition performance reported on Kaggle.

Index Terms— Audio tagging, Convolutional neural networks, Model ensemble, Transfer Learning

1. INTRODUCTION

The development of broadly-applicable sound event classification models that consider an increased and diverse amount of categories is highly demanded. These models can be used, for example, in automatic description of multimedia or acoustic monitoring applications.

Deep learning techniques have been widely used in the automatic audio tagging community as they can automatically exploit useful features for audio classification. Besides, ensembling diverse deep learning models is essential for developing accurate automatic audio tagging systems, as different architectures are in general able to exploit different aspects of the audio inputs.

For the automatic audio tagging task, the input features of deep learning models are of great importance. Though deep learning models are known for their ability to automatically extract meaningful features from the input data (i.e. using 1d convolution to directly process raw audio waveforms), the audio tagging community has shown that calculating the spectrogram of audio is still necessary to achieve high recognition performance [1].

Deep learning models usually require large amounts of training data to achieve high recognition performance. For example, large image datasets, such as ImageNet [2], that contain millions of labelled images make it possible to train a very deep model for accurate image classification [3]. However, for some domain-specific tasks like medical image classification, it is unrealistic to construct a dataset as large as ImageNet. Thus, transfer leaning is often employed for domain-specific tasks to accelerate the training process and improve the final performance.

In this work, we ensemble four deep learning models for automatic audio tagging. Two models are based on 2D CNNs and another two are based on 1D CNNs. 2D CNNs models respectively take Mel spectrogram and MFCC spectrogram as input features, while 1D CNNs models directly process raw audio waveforms. Our models are finetuned from two external datasets that are larger than FreeSound [4]: SoundNet [5] and VEGAS [6]. These components, together with intensive data augmentation techniques and well-optimized training strategies, contribute to our best audio tagging performance reported on Kaggle.

2. METHODS

2.1. Data Preprocessing

The audios provided by FreeSound are in different lengths (from less than two seconds to more than 10 seconds). However, CNNs with fully connected layers can only process fix-sized input data. To handle this issue, audios longer than 2 seconds are randomly cropped to 2s length while those shorter than 2s are zero-padded to 2s length during training and testing. Although it is common to resample the input audios to a lower sampling rate, we don't perform this operation in our system, as we find that it leads to interior performance in general.

The raw audios of 2s are then fed to the 1D CNN models for training. Meanwhile, Spectrograms are calculated in order to train the 2D CNN models. This is done by first performing STFT with window size of 2048 and hop length of 512. Then Mel filterbank or MFCC filterbank are applied to the STFT with 80 filters. As a result, we obtain spectrograms of size 173 × 80 for audios of 2s.

2.2. Model Architecture

2.2.1. 2D CNNs

We compared three 2D convolution architectures (CNNs without short-cut connections, ResNet [7] and DenseNet [8]) and find that DenseNet is the best-performing model. Our DenseNet model contains five dense blocks, with four "bn-relu-conv" (batch normalization, relu activation, 2d convolution layer) units in each dense block. The growth rate k is set to 32. Global max pooling is used at the end of DenseNet to reduce the time and the frequency axes. Kernel regularization (1×10^{-4}) is applied to all convolution and fully connected layers. The kernel sizes of all convolution layers are 3×3.

2.2.2. 1D CNNs

Our 1D CNNs directly process raw audio waveforms. Two different 1D convolution models are adopted.

The first model is the fine-tuned version of the pre-trained SoundNet model [5]. We refer the reader to [5] for the detailed model architecture.

The second model has ten residual connection blocks [7], with two 1D convolution layers in each block. 1D Max pooling is used between every two residual blocks to halve the feature map size along the time axis. Every time when the feature map size is halved, the feature map channels double. 1D global max pooling is used at the end of the model to reduce the time axis. The kernel size of the first two 1D convolution layers is 32, while the kernel size of the following convolution layers is 3. Each max pooling layer is followed by a dropout layer with dropout rate 0.1.

2.3. Training Strategies

All of our models are trained using Keras [9]. Adam [10] with initial learning rate euqals 0.001 is selected as the optimizer. The learning rate is reduced by 10 times when the validation loss has not decreased for 5 epochs. Besides, cyclical learning rate (CLR) [11] is used to provide quicker model converge. We adopt the triangular CLR policy and set the number of training iterations per half cycle to the number of batches in each epoch.

1D CNNs (SoundNet) are finetuned from the 1D CNNs in [5], while 1D CNNs (ResNet), 2D CNNs (Mel Power Spectrogram) and 2D CNNs (MFCC) are pretrained using VEGAS dataset [6],then fintuned with the Freesound dataset.

In addition, we apply various data augmentation techniques to the raw waveforms before converting them to spectrograms, including time stretching, pitch shifting, reverb and dynamic range compression. These operations are implemented using SoX[1].

We use 10-fold cross validation during training. For each model, the predicted probabilities of 10 folds on the test set are averaged to give the final result.

2.4. Evaluation Strategies

We zero-pad or crop randomly the test audios to 2s before predicting their classes with the trained models. However, it is possible that the randomly cropped audio frame may not contain sufficient information for classification (e.g. a silent 2s frame may be chosen out of an original audio labelled as ”bark”). To alleviate this issue, we discard the prediction result in case it is not certain enough (in the sense that the highest output probability is smaller than 0.7) and randomly choose another 2s portion of the test audio for classification again.

3. EXPERIMENTAL RESULTS

In this section, we report the mean top-3 accuracy of each model on the test set as shown in Table 1. It can seen that 2D CNNs are superior to 1D CNNs; on the other hand, Log-Mel spectrogram is better than MFCC spectrogram (1D convolution) for this specific task. Finally, we ensemble these four models by geometrically averaging their prediction probabilities to get the submission result. The weights of each model are chosen based on their respective performance and shown in Table 1.

Model	MAP@3	Weight
1D Conv (SoundNet)	0.874 ± 0.006	0.2
1D Conv (ResNet)	0.887 ± 0.010	0.2
2D Conv (DenseNet, MFCC)	0.924 ± 0.006	0.2
2D Conv (DenseNet, Log-Mel)	0.938 ± 0.003	0.4

Table 1: Average top-3 accuracy and ensemble weight of each model calculated on the validation folds. 10-fold cross validation is used.

4. REFERENCES

[1] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, “Cnn architectures for large-scale audio classification,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. [Online]. Available: https://arxiv.org/abs/1609.09430

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 248–255.

[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] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, “Freesound datasets: a platform for the creation of open audio datasets,” in *Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93.* International Society for Music Information Retrieval (ISMIR), 2017.

[5] Y. Aytar, C. Vondrick, and A. Torralba, “Soundnet: Learning sound representations from unlabeled video,” in *Advances in Neural Information Processing Systems*, 2016, pp. 892–900.

[6] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg, “Visual to sound: Generating natural sound for videos in the wild,” *arXiv preprint arXiv:1712.01393*, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks.” in *CVPR*, vol. 1, no. 2, 2017, p. 3.

[9] F. Chollet *et al.*, “Keras: Deep learning library for theano and tensorflow,” *URL: https://keras. io/k*, vol. 7, no. 8, 2015.

[10] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[11] L. N. Smith, “Cyclical learning rates for training neural networks,” in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 2017, pp. 464–472.

[1] http://sox.sourceforge.net/