BUPT SUBMISSIONS TO DCASE 2020: LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION WITH POST TRAINING STATIC QUANTIZATION AND PRUNE

Technical Report

Jiawang Zhang, Chunxia Ren, Shengchen Li

Beijing University of Posts and Telecommunications, Beijing, P. R. China
(zhangjiawang, chunxiaren, shengchen.li)@bupt.edu.cn

ABSTRACT

This report describes a method for Task 1b (Low-Complexity Acoustic Scene Classification) of the DCASE 2020 challenge, which targets low complexity solutions for the classification problem. The proposed model has five residual block with average pooling. To improve the performance of the proposed system, binaural features from the dataset are used, and with Log Mel Spectrogram, mix-up data augmentation. To reduce system complexity, the proposed method uses Post Training Static Quantization and Prune methods, Post Training Static Quantization are used to do the 8-bits quantization, this method can reduce the model size by four times. Pruning can reduce redundant weights by prune the low weights, the process allows only a small part of the original weight parameters performance close to the original network. The accuracy of the method proposed in this report on the development data set is 92.9%, which is 5.6% higher than the baseline, but 81% lower than the baseline model.

Index Terms— Low-Complexity Acoustic Scene Classification, ResNet, Post Training Static Quantization, Prune.

1. INTRODUCTION

In daily life, sound carries a lot of information, which can be used to judge the surrounding scene (acoustic scene). In recent years, deep neural networks have out performance other systems in many machine learning tasks [1]. Including acoustic scene classification and acoustic event recognition. The performance of computer in such tasks is constantly improving, even better than the human ear recognition [2]. The development of algorithms that use computers to automatically extract acoustic scene information has great potential in a variety of applications. For example, searching multimedia based on audio content, making context-aware mobile devices, intelligent monitoring systems that can sense hearing, robot hearing, unmanned cars.

Queen Mary University of London (QMUL) organized the first DCASE (Challenge on Detection and Classification of Acoustic Scenes and Events) challenge in 2013 [3], paving the way for sound detection classifier performance evaluation. The DCASE Challenge aims to expand the most advanced technical sound scene and event analysis methods,

In this year's DCASE2020 challenge, Task1b targets low complexity solutions for the classification problem in term of model size for the first time.

The models of deep neural network need a large number of weights [4], hence need large storage and memory bandwidth. This makes it difficult to deploy these models on mobile systems and embedded systems which have limited hardware resources. Firstly, different applications are updated through different app stores for many mobile-first companies, and these applications are very sensitive to the size of the binary files. Therefore, the feature of increasing the binary size by 100MB will be more scrutinized than the feature of increasing it by 10MB. Less network bandwidth, real-time processing, and greater storage overhead make deep neural networks difficult to integrate into mobile applications. Secondly, many memory bandwidths to get weights is required to running large neural networks, and a lot of calculations to do dot products which in turn consumes a lot of energy. Mobile devices have limited battery capacity, so power hungry applications such as deep neural networks are difficult to deploy.

So to run inference of these large networks on mobile devices and embedded systems, the requirements of the storage and energy requirements have been reduced [5].

Neural network has redundancy parameterization. There is significant redundancy in neural networks of many depths. In fact, Denil and his team, has shown that using only a small part (5%) of the weights is sufficient to predict the remaining weights [6]. The paper also proposes that these remaining weights can even be left unattended. That is to say, training only a small part of the original weight parameters may reach the performance of the original network similar to or even exceed the original network.

In this report, in order to improve the classification accuracy of the system, our method extracts binaural features from the dataset, and uses Log Mel Spectrogram to process features, then using Mix-up to do the data augmentation [7]. To compress system complexity, this report use Post Training Static Quantization and Prune methods. Post Training Static Quantization and Prune is used to slim the models. Post Training Static Quantization are used to do the 8-bits quantization aim at reduce the model size by four times [8]. Prune proposed to trim the channels with smaller input weights in trained ResNet, and then fine-tune the network to restore accuracy [9]. This method can remove some redundant parameters and reduce the size of the network model when the accuracy loss is very small. Experiments show that the accuracy of this report is 5.6% higher than baseline, and the model complexity is lower than baseline.

2. DATA PREPROCESSING

This section describes the methods about signal processing and data augmentation that this report to transform the audio sample into acoustic features.

2.1. Acoustic Feature

The audio in the dataset is binaural audio samples, so this report extracts the left and right channels to process the audio. And this report use Log Mel spectrogram as audio feature. When extracting features from audio, Log Mel spectrogram is generally used compared to MFCC, wavelet, Log Mel has better performance [10].

The audio of the dataset is provided in binaural, 48000Hz 24-bit format. Every audio sample is firstly resampled to 44100Hz, this report use 2018 sample windows and the hop-size of 1536 sample to divided an audio sample into 280 frames. This report use 2 channels and 256 mel_bins, so the Log Mel spectrogram in the shape of (280, 256, 2).

2.2. Data Augmentation

This report use Mix-up to enhance the data. Mix-up is an unconventional data enhancement method, a simple data enhancement principle that has nothing to do with data [11]. It uses linear interpolation to construct new training samples and labels. The final treatment of the label is shown in the following formula:

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j \quad (1)$$

$$\tilde{y} = \lambda y_i + (1-\lambda)y_j \quad (2)$$

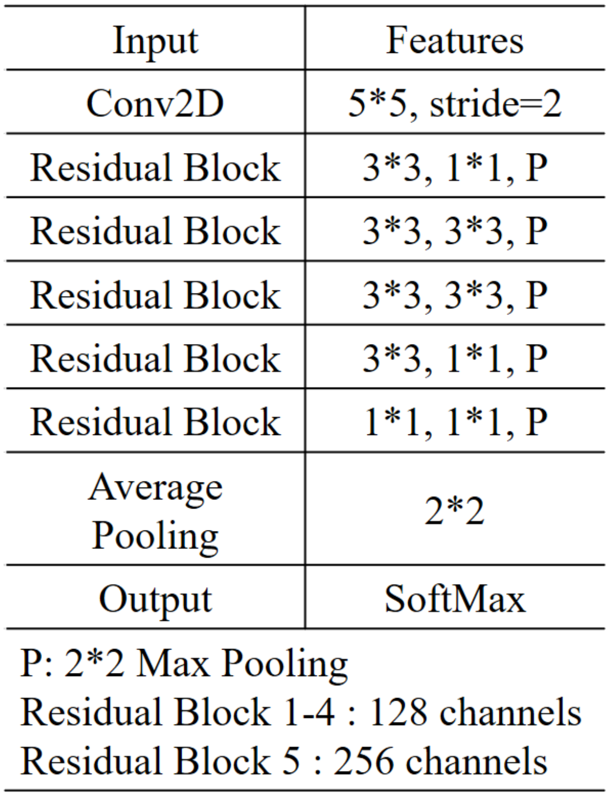
(x_i, y_i) and (x_j, y_j) are two samples randomly selected from the training data. where x_i, x_j are raw input vectors, and y_i, y_j are one-hot label encodings. The λ follows the beta distribution. $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$. As α increases, the training error of the network will increase, and its generalization ability will increase accordingly. And when $\alpha \to \infty$, the model will degenerate into the original training strategy.

3. METHOD

This section introduces the methods used to reduce the network model. This report use ResNet model [12], extracts the two channels to process the audio, uses Log-Mel spectrogram as audio feature. Mix-up is used to do data augmentation. Post Training Static Quantization and Prune methods are used to compress system complexity.

3.1. Network Structure

The network we used was ResNet network proposed by Khaled, the input of the network is 5*5, the stride is 2, and then has five residual block and an average pooling, the output to do the SoftMax [13]. Residual block 1-4 is 128 channels, residual block 5 is 256 channels, with 2*2 max pooling.



Input	Features
Conv2D	5*5, stride=2
Residual Block	3*3, 1*1, P
Residual Block	3*3, 3*3, P
Residual Block	3*3, 3*3, P
Residual Block	3*3, 1*1, P
Residual Block	1*1, 1*1, P
Average Pooling	2*2
Output	SoftMax

P: 2*2 Max Pooling
Residual Block 1-4 : 128 channels
Residual Block 5 : 256 channels

Figure 1: Network Structure of ResNet

3.2. Post Training Static Quantization

Quantization [14] usually uses this method, this method quantifies the weight in advance, Post Training quantization is usually used for CNN [15]. The process of post training static quantification is: First, prepare the model and specify the activation values to be quantized and dequantized. The model cannot be reused. Convert operations that need to be quantified again into modules. Then fuse the operations and choose the configuration of the quantization methods. Finally use Pytorch function to quantify the model.

Post Training Static Quantization are used to do the 8-bits quantization, this method can reduce the model size by four times. This model's total size is 83.46KB, and baseline's model total size is 450KB, model size of this report is much smaller than the baseline model size.

3.3. Pruning of network

This report also uses Prune to reduce the model size, which proposes to trim the channels with smaller input weights in trained ResNet, and then fine-tune the network to restore accuracy [16]. Before training, introduce sparsity by randomly deactivating the input-output channel connections on the convolutional layer, which also produces a smaller network with a moderate loss of accuracy.

For each channel, a scaling factor γ is introduced and then multiplied by the output of the channel. Then jointly train the network weights and these scaling factors, and finally remove the channels with small scaling factors directly to fine-tune the pruned network. In particular, the objective function is defined as:

$$\mathrm{L} = \sum_{(x,y)} l(f(x,W),y) + \lambda \sum_{\gamma\in\Gamma} g(\gamma) \qquad (3)$$

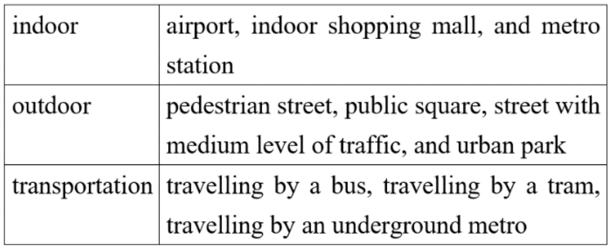
Where (x,y) represents training data and labels, W is a trainable parameter of the network, and the first term is the training loss function of CNN. $\mathrm{g}(.)$ is the multiplication term on the scaling factor, and λ is the balance factor of the two terms. In the experiment process, $\mathrm{g}(\mathrm{s}) = |\mathrm{s}|$, which is L1 regularization, is also widely used in sparseness.

The output of each layer of ResNet will be used as the input of subsequent layers, and batch normalization layer is before the convolution layer. In this case, the thinning is obtained at the input end of the layer, and one layer selectively accepts all the subset of channels is used for the next convolution operation. In order to save parameters and running time during testing, a channel selection layer needs to be placed to identify important channels. Then retrain the model, can get a low complexity network.

4. EXPERIMENTS AND RESULTS

4.1. Audio Dataset

The dataset that this report uses is TAU Urban Acoustic Scenes 2020 3 Class [17]. The development dataset was collected by Tampere University of Technology. The total amount of audio in the development set is 40 hours, it contains ten different acoustic scenes form 10 European cities. The ten acoustic scenes are grouped into three major classes, they are Indoor scenes, Outdoor sense and Transportation related scenes. The development dataset has been divided into 14400 segments, there are 9185 segments in the training part, and 4185 segments in the testing part. The evaluation set contain data from 12 cities (2 cities unseen in the development set). Evaluation data contains 30 hours of audio.



indoor	airport, indoor shopping mall, and metro station
outdoor	pedestrian street, public square, street with medium level of traffic, and urban park
transportation	travelling by a bus, travelling by a tram, travelling by an underground metro

Figure 2: Three classes of the dataset

4.2. Baseline Result

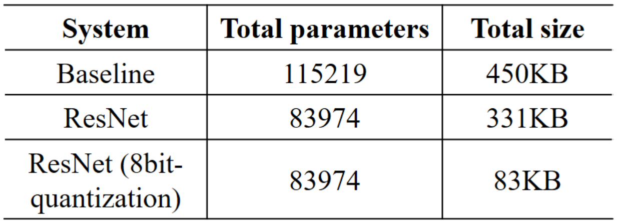
In subtask B, the baseline system is similar to DCASE2019 baseline. The system implements a convolutional neural network (CNN) based approach. Log Mel-band energies are first extracted for each 10-second signal, and a network consisting of two CNN layers and one fully connected layer is trained to assign scene labels to the audio signals. The model size of the system is 450 KB.

4.3. Task1b Result

This section compares the baseline model and this report model.

4.3.1. Model Size

Compared with baseline model, the model of this report is much smaller than the model of baseline, which saves more memory. The total size of the baseline is 450KB, the total size of the ResNet is 331KB before quantization, after 8bit-quantization, the total size is 83KB.

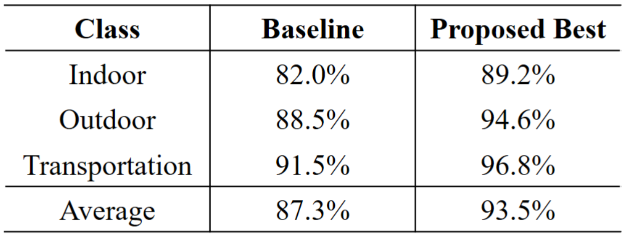


System	Total parameters	Total size
Baseline	115219	450KB
ResNet	83974	331KB
ResNet (8bit-quantization)	83974	83KB

Figure 3: Total size of the models

4.3.2. Accuracy

The accuracy of the method proposed in this report on the development data set is 92.9%, which is 5.6% higher than the baseline. The accuracy of transportation is best. and the accuracy of the indoor is relatively lower than others.



Class	Baseline	Proposed Best
Indoor	82.0%	89.2%
Outdoor	88.5%	94.6%
Transportation	91.5%	96.8%
Average	87.3%	93.5%

Figure 4: Accuracy of baseline and proposed best

5. CONCLUSION

This report concludes the methods for DCASE task1b. This report use ResNet model, extracts the two channels to process the audio, uses Log-Mel spectrogram as audio feature. Mix-up is used to do data augmentation. By use Post Training Static Quantization and Prune methods can reduce the model size, and almost no loss of accuracy. This Low-Complexity Acoustic Scene Classification is suit for mobile systems and embedded systems which have limited hardware resources.

6. REFERENCES

[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C]//Advances in neural information processing systems. 2012: 1097-1105.

[2] Barchiesi D, Giannoulis D, Stowell D, et al. Acoustic scene classification: Classifying environments from the sounds they produce [J]. IEEE Signal Processing Magazine, 2015, 32(3): 16-34.

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

[4] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv:1409.1556, 2014. A. B. Smith, C. D. Jones, and E. F. Roberts, “A sample paper in journals,” *IEEE Trans. Signal Process.*, vol. 62, pp. 291-294, Jan. 2000.

[5] Lee E A, Seshia S A. Introduction to embedded systems: A cyber-physical systems approach [M]. Mit Press, 2016.

[6] Denil M, Shakibi B, Dinh L, et al. Predicting parameters in deep learning [C]//Advances in neural information processing systems. 2013: 2148-2156.

[7] Shen J, Pang R, Weiss R J, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions [C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 4779-4783.

[8] https://pytorch.org/docs/stable/quantization.html

[9] Liu Z, Li J, Shen Z, et al. Learning efficient convolutional networks through network slimming [C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 2736-2744.

[10] Muda L, Begam M, Elamvazuthi I. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques [J]. arXiv preprint arXiv:1003.4083, 2010.

[11] Zhang H, Cisse M, Dauphin Y N, et al. mixup: Beyond empirical risk minimization [J]. arXiv preprint arXiv:1710.09412, 2017.

[12] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning [C]//Thirty-first AAAI conference on artificial intelligence. 2017.

[13] Koutini K, Eghbal-Zadeh H, Dorfer M, et al. The Receptive Field as a Regularizer in Deep Convolutional Neural Networks for Acoustic Scene Classification [C]//2019 27th European Signal Processing Conference (EUSIPCO). IEEE, 2019: 1-5.

[14] Gray R M, Neuhoff D L. Quantization [J]. IEEE transactions on information theory, 1998, 44(6): 2325-2383.

[15] Xu Y, Kong Q, Wang W, et al. Large-scale weakly supervised audio classification using gated convolutional neural network [C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 121-125.

[16] Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding [J]. arXiv preprint arXiv:1510.00149, 2015.

[17] Mesaros A, Heittola T, Virtanen T. A multi-device dataset for urban acoustic scene classification [J]. arXiv preprint arXiv:1807.09840, 2018.