Low-Complexity Acoustic Scene Classification Using Data Generation Based On Primary Ambient Extraction

Chuang Shi, Haocong Yang, Yingzi Liu, Jiangnan Liang School of Information and Communication Engineering

University of Electronic Science and Technology of China, Chengdu, China

Abstract—Acoustic scene classification (ASC) is an important branch of machine hearing. Since ASC systems are intended to be deployed on mobile devices, how to ensure the performance under low-complexity implementation has become an attracting research problem. The state-of-the-art methods include compressing parameter precisions, reducing quantization bits, introducing sparsity constraints and so on. These methods mainly focus on the model level optimization, while explorations are rarely originated from the data level. This paper introduces a train of thoughts from data level, inspired by a stereo audio processing algorithm, namely the primary ambient extraction (PAE), which generates additional samples through audio up-mixing. The experiment results demonstrate that the proposed method exhibits better performance than a group of ASC baseline systems without data level optimization, not to mention that the proposed method is compatible with the existing model level optimization.

Index Terms—Acoustic scene classification, low-complexity implementation, convolutional neural network, primary ambient extraction

I. INTRODUCTION

In recent years, resultant from the rapid development of artificial intelligence, more and more new market demands have emerged. As one of the main ways of human computer interaction, machine hearing has been attracting extensive attention from academia and industry. ASC is an important branch of machine hearing, which aims to judge the environment of sound transmission through recorded audio signals. ASC systems can be deployed in the fields of disability assistance, autonomous driving, and multimedia material archiving.

According to previous subjective testing results, the performance of machine hearing in ASC has far exceeded that of human hearing [1]. Human beings are not naturally good at distinguishing acoustic scenes. Peltonen et al. demonstrate that people's perception of acoustic scenes relies on cognition of typical sound events, and people lose their ability to judge when facing acoustic scenes that are weakly correlated with typical sound events [2].

However, ASC is facing implementation difficulties. ASC is intended to be deployed in mobile devices in most application scenarios, where communication and computing capacities are rather limited [3]. Convolutional neural network (CNN) models can recognize scaling, displacement and other 2dimensional distortion invariance [4]. They are the mainstream choice for ASC system implementation [1]. However, a CNN



Fig. 1. Approaches of parameter precision compression.

model often consists of an overwhelming number of parameters. Therefore, how to ensure the performance of the ASC system in low complexity, more precisely, how to optimize the ASC system under a relatively low level of parameter storage capacity, has become an emerging research problem.

At this stage, there are several well-known strategies for ASC system implementation with low level of parameter storage capacity. The compression of parameters precision can accommodate more parameters with the same memory space. Compressing FLOAT32 into FLOAT16 is one of the most commonly adopted methods. When dealing with a small number of classes, this method has achieved considerable performance [5] [6]. On this basis, quantization is also a feasible idea. Hu et al. use INT8 quantization for parameters to further compress the memory usage of a single parameter [7], while McDonnell et al. use the extreme quantization method of just 1 bit per parameter [8], which increases the parameter amount by an order of magnitude compared with the conventional model using FLOAT32. The comparison of the above methods is shown in Fig. 1.

In addition, Chang el al. propose to prune the parameters whose weight is lower than a threshold value, in order for the parameters of the deep learning model to possess sparsity [9]. Koutini et al. use the decomposed convolutional layer, which is inspired by the singular value decomposition [10] [11]. Through the split and reorganization of the convolution layers, the model scale is directly saved to one eighth of the original size.

The state-of-the-art methods mainly focus on the model level optimization. Previous works have demonstrated that there is still room for exploration from the data level [12] [13]. In this paper, a stereo audio processing algorithm, namely the PAE, is introduced to generate additional audio samples through up-mixing. As a data level optimization, the PAE can be integrated with existing model compression methods. A fast implementation of the PAE algorithm is also proposed to reduce its complexity. Experiments are carried out on the TAU urban acoustic scenes 2020 3-class development dataset. The proposed method is compared with and outperforms a group of ASC baseline systems without data level optimization.

II. PROPOSED APPROACH

The PAE algorithm was originally proposed to up-mix a stereo audio clip into an arbitrary number of channels in order to be played back by multi-channel reproduction systems. It assumes that in every channel of a stereo audio, there is a primary component and an ambient component, which are written as

$$x_c(t) = p_c(t) + a_c(t), \quad \forall c \in \{0, 1\},$$
(1)

where $c \in \{0, 1\}$ is the channel index. The primary components p_c are assumed to be correlated with each other and only different in the amplitude with a panning factor k, i.e. $p_1 = kp_0$. The ambient components a_c are assumed to have the same energy but uncorrelated with each other. The ambient components are also uncorrelated with the primary components. These are crucial spatial assumptions in the derivation of the PAE algorithm [14].

After the short-time Fourier transform (STFT), (1) is rewritten as

$$\mathbf{X}_{c}[m,f] = \mathbf{P}_{c}[m,f] + \mathbf{A}_{c}[m,f], \quad \forall c \in \{0,1\}, \quad (2)$$

where m is the index of frame and f is the index of frequency bin. The notations [m, f] are omitted for brevity in the latter part of this paper.

The spectra of ambient components are expressed as

$$\mathbf{A}_{c} = |\mathbf{A}_{c}| \odot \mathbf{W}_{c}, \quad \forall c \in \{0, 1\},$$
(3)

where \mathbf{W}_c is $\mathbf{W}_c(m, f) = e^{j\theta_c(m, f)}$ and $\theta_c(m, f)$ is the element of θ_c in the time-frequency bin (m, f). $\theta_c = \angle \mathbf{A}_c$ is the vector of phase angles of the ambient components.

Since $P_I = k P_0$, (2) leads to

$$\mathbf{X}_1 - k\mathbf{X}_0 = \mathbf{A}_1 - k\mathbf{A}_0. \tag{4}$$

Substituting (4) into (3) yields

$$|\mathbf{A}| = (\mathbf{X}_1 - k\mathbf{X}_0) / (\mathbf{W}_1 - k\mathbf{W}_0).$$
 (5)

Let the phase angle of $(\mathbf{X}_1 - k\mathbf{X}_0)$ be $\boldsymbol{\theta}$. Since $|\mathbf{A}|$ is real, $\sin \theta / \cos \theta = (\sin \theta_1 - k \sin \theta_0) / (\cos \theta_1 - k \cos \theta_0)$ must hold. It is further manipulated as

$$\sin\left(\boldsymbol{\theta} - \boldsymbol{\theta}_0\right) = k^{-1}\sin\left(\boldsymbol{\theta} - \boldsymbol{\theta}_1\right). \tag{6}$$

There are two tentative solutions of θ_0 . They are

$$\boldsymbol{\theta}_{0}^{(0)} = \boldsymbol{\theta} - \boldsymbol{\alpha}, \\ \boldsymbol{\theta}_{0}^{(1)} = \boldsymbol{\theta} + \boldsymbol{\alpha} + \boldsymbol{\pi},$$
 (7)

where $\boldsymbol{\alpha} = \arcsin \left[k^{-1} \sin \left(\boldsymbol{\theta} - \boldsymbol{\theta}_1 \right) \right]$ and $\boldsymbol{\alpha} \in \left[-0.5\pi, 0.5\pi \right]$. Moreover, the imaginary part of $(\mathbf{W}_1 - k\mathbf{W}_0)$ has a same sign with that of $(\mathbf{X}_1 - k\mathbf{X}_0)$. This leads to $\left(\operatorname{Im} \{ \mathbf{W}_1 - k\mathbf{W}_0 \} / \operatorname{Im} \{ \mathbf{X}_1 - k\mathbf{X}_0 \} \right) |_{\boldsymbol{\theta}_0} \geq 0$. Therefore, $\boldsymbol{\theta}_0 = \boldsymbol{\theta} + \boldsymbol{\alpha} + \boldsymbol{\pi}$ is the only solution to (6).

By substituting (3) and (5) into (2), we have

$$\mathbf{A}_{c} = \left(\mathbf{X}_{1} - k\mathbf{X}_{0}\right) / \left(\mathbf{W}_{1} - k\mathbf{W}_{0}\right) \odot \mathbf{W}_{c}, \mathbf{P}_{c} = \mathbf{X}_{c} - \left(\mathbf{X}_{1} - k\mathbf{X}_{0}\right) / \left(\mathbf{W}_{1} - k\mathbf{W}_{0}\right) \odot \mathbf{W}_{c}, \qquad (8) \forall c \in \{0, 1\},$$

where \mathbf{X}_c and k can be computed from the correlation of input signals. Up to this stage, $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are still unknown. Due to the spatial assumptions between $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$, only one phase angle $\boldsymbol{\theta}_1$ needs to be estimated. With a sparsity constraint, the PAE is transformed into an optimization problem, which is expressed as

$$\hat{\boldsymbol{\theta}}_{1}^{*} = \arg\min_{\hat{\boldsymbol{\theta}}_{1}} \left\| \hat{\mathbf{P}}_{1} \right\|_{1}.$$
(9)

The standard implementation of the PAE algorithm is based on a time-consuming angle-by-angle searching strategy [15]. Instead, an approximated solution to θ_1 can be analytically obtained [16]. According to (8), the approximated solution is given by

$$\hat{\boldsymbol{\theta}}_{\text{les}}^{*} = \begin{cases} \boldsymbol{\angle} \mathbf{X}_{1}, \forall k > 1; \\ \boldsymbol{\angle} \left(\mathbf{X}_{1} - \mathbf{X}_{0} \right), \forall k = 1. \end{cases}$$
(10)

This approximated solution often causes a notable loss of accuracy [17]. Therefore, a fast implementation of the PAE algorithm is proposed by optimizing the searching strategy. The range of angles are limited to be centered at the approximated solution with an offset not exceeding $\pm\beta$. The proposed solution to θ_1 can be expressed as

$$\hat{\boldsymbol{\theta}}_{1}^{*} = \begin{cases} \arg \min_{\substack{\boldsymbol{\theta} = [\angle X_{1} - \beta, \angle X_{1} + \beta]}} \left\| \hat{\mathbf{P}}_{1}(\boldsymbol{\theta}) \right\|_{1}, \forall k > 1; \\ \arg \min_{\substack{\boldsymbol{\theta} = [\angle (X_{1} - X_{0}) - \beta, \angle (X_{1} - X_{0}) + \beta]}} \left\| \hat{\mathbf{P}}_{1}(\boldsymbol{\theta}) \right\|_{1}, \forall k = 1 \end{cases}$$
(11)

This results in a fast implementation and equivalent accuracy as compared to the original PAE algorithm.

The PAE algorithm is thereafter applied to each stereo audio sample to extract the primary and ambient components of the left and right channels. However, considering that the primary and ambient components are not with the same data distribution as the original stereo audio, it is not suggested to use them as individual samples. Therefore, as shown in Fig. 2, during the process of each iteration, the primary and ambient components are remixed according to a random weight, which can provide augmented acoustic features that the conventional mix-up method cannot provide. Due to the correlation between the augmented samples and the original stereo audio sample, the proposed method can increase the generalization of data without modifying the original data distribution.



Fig. 2. Data generation using the PAE algorithm.

III. EXPERIMENTS

Experiments are conducted on the TAU urban acoustic scenes 2020 3-class development dataset, which is a highquality binaural recorded dataset and consists of various acoustic scene samples collected in 10 cities of Europe [3]. 40 hours of recordings are divided into 14400 segments, with 10 seconds length for each segment. Considering the learning ability of the model under low complexity is limited, the dataset is only divided into 3 classes. They are the classes of indoor, outdoor, and transportation. The development dataset is split into the training subset and the evaluation subset upon the release.

A. Fast Implementation of the PAE Algorithm

Compared with the previous work [15], the PAE algorithm is optimized in two aspects to achieve low complexity. Firstly, the acoustic features are transformed from log-mel energies to constant-Q transform (CQT). The former uses STFT first and then passes through mel filter bank, while the latter only needs CQT directly. The PAE algorithm is implemented after STFT or CQT. The number of CQT calculation points in each frame is close to the log-mel energies, which is far lower than STFT. Therefore, the total number of PAE calculation points can be greatly reduced.

In addition, the previous PAE implementation searches phase angles of the whole 360-degree range in each timefrequency bin. As shown in Section 3, this process can be improved by limiting the search range of angles to $\pm 20^{\circ}$ of the approximated analytical solution with almost no performance degradation. The time costs for processing 14400 audio samples are shown in Fig. 3. "Raw" denotes the conventional preprocessing without any up-mixing methods. "Fast PAE" denotes the proposed fast implementation, while "PAE" denotes the standard implementation. The time costs of different processing methods are recorded by the same personal computer equipped with Intel i5-9600K processor running at 5.00GHz. Since the time costs of model training in all experiments are less than half an hour, the improvement



Fig. 3. Time costs of processing 14400 audio samples.

achieved by the fast implementation of the PAE algorithm is crucial.

B. Data Generation

Every audio sample with the length of 10 seconds is firstly resampled to 22.05kHz and divided into 216 frames with 1024point Hanning windows. In the processing of CQT, $C0 \approx$ 16.35 Hz is set as the lowest tone. There are 336 bins in total and every 36 bins represent one octave. Finally, the energy spectrum of CQT is transformed to the decibel scale. Acoustic features with the shape of (2, 336, 216) are obtained at last. The first dimension of the acoustic features indicates that there are 2 channels, the left and right channels, in each stereo audio sample.

A full CNN is adopted as the ASC model. Residual blocks are further included in the structure, which is shown in Table. I [18]. The parameter precision is compressed to FLOAT16 to reduce the model size. During the training of the CNN model, the min-max method is used to treat the model input in order for the convergence to be accelerated and the numerical

TABLE I

THE STRUCTURE OF MODEL, WHERE THE KSIZE1 AND KSIZE2 DENOTE THE SIZE OF THE FIRST AND SECOND CONVOLUTION LAYERS OF A RESIDUAL BLOCK, RESPECTIVELY.

Input 2×336×216
Conv2d (ksize=5, pading=2, stride=2, channel=48)
BatchNorm2d(channel=48)
ReLU(channel=48)
ResidualBlock (ksize1=3, ksize2=1, channel=48)
MaxPooling (size=2)
ResidualBlock (ksize1=3, ksize2=1, channel=48)
MaxPooling (size=2)
ResidualBlock (ksize1=3, ksize2=1, channel=48)
MaxPooling (size=2)
ResidualBlock (ksize1=1, ksize2=1, channel=96)
ResidualBlock (ksize1=1, ksize2=1, channel=96)
ResidualBlock (ksize1=1, ksize2=1, channel=96)
Conv2d (ksize=1, padding=2, stride=2, channel=3)
BatchNorm2d (channel=3)
GlobalAveragePooling2d (channel=3)
Output 3-way SoftMax

problems to be avoided. The adaptive moment estimation algorithm is chosen as the optimizer, where the learning rate, betas, eps, and mini batch size are set to 0.001, (0.9, 0.999), 2^{-11} , and 32, respectively. The mix-up method is also adopted with the hyper-parameter alpha of 0.2 [19]. The above settings are applied on PyTorch 1.5.1 and Nvidia GeForce RTX 2080Ti (11GB).

Furthermore, common model level optimizations are performed on the proposed model. In addition to the use of FLOAT16 as the default parameters precision setting, decomposing convolution layers and enhancing parameter sparsity are also involved. When decomposing the convolution layers, the compression parameter is set to Z = 4 [11]. Pruning is carried out by the L1 sparsity constraint and adopting pruning rates of 0.1 and 0.2 [9].

The experiment results of low-complexity ASC systems are shown in Table. II, where training and evaluation are carried out 10 times for each method such that the mean and deviation of the classification accuracy can be calculated. Another group of results with the the PAE-based data augmentation are shown in Table. III. "CNN+log-mel (Baseline)" refers to the baseline system provided by the dataset [3], which is a conventional CNN model using acoustic features of log-mel energies. "CNN+CQT" denotes the baseline CNN model structure using acoustic features extracted by CQT. "ResNet+CQT" refers to the ResNet model (shown in Table. I) using acoustic features extracted by CQT. "Decomp", "Prune", and "FP" refers to the actions of decomposing convolution layers, adopting *L*1 constrained unstructured pruning, and the proposed fast implementation of the PAE algorithm, respectively.

The first two rows of Table. II and Table. III show the performance improvement coming from the optimization of the acoustic feature and model structure. The rest of the results are obtained by the same model structure to quantify the impact of different methods. Decomposing convolution layers can reduce the parameter scale by more than half,

 TABLE II

 The experimental results of low-complexity ASC systems.

model	Total non-zero parameters	Model Size	Macro-average accuracy (%)
CNN+log-mel (Baseline) [3]	110579	450KB (float32)	87.3 (± 0.7)
CNN+CQT	110579	450KB (float32)	92.0 (± 0.4)
ResNet+CQT	210438	456KB (float16)	95.4 (± 0.3)
ResNet+CQT (Decomp)	91930	365KB (float16)	94.5((± 0.3)
ResNet+CQT (Prune0.1)	189395	410KB (float16)	95.3(± 0.6)
ResNet+CQT (Prune0.2)	168360	365KB (float16)	94.0 (± 1.4)

TABLE III THE EXPERIMENTAL RESULTS OF LOW-COMPLEXITY ASC SYSTEMS WITH THE PAE-BASED DATA AUGMENTATION.

model	Total non-zero parameters	Model Size	Macro-average accuracy (%)
CNN+log-mel	110579	450KB	87.3 (± 0.7)
(Baseline)		(float32)	,
CNN+CQT	110579	450KB	$92.4 (\pm 0.3)$
(FP)		(float32)	
ResNet+CQT	210438	456KB	96.0 (± 0.2)
(FP)		(float16)	
ResNet+CQT	91930	237KB	95.0(+0.3)
(Decomp+FP)		(float16)	99.0(± 0.5)
ResNet+CQT	189395	410KB	95.8(± 0.5)
(Prune0.1+FP)		(float16)	
ResNet+CQT	168360	365KB	94.4(± 1.5)
(Prune0.2+FP)		(float16)	

while achieving the performance loss of just 1%. Pruning introduces sparsity to the parameters and increases the uncertainty of model performance. There is few performance loss when the pruning rate is within a threshold range, but the performance will drop rapidly after the threshold is exceeded. The pruning rate at 0.2 in Table. III is more uncertain than that in Table. II. PAE enhances the resistance of the model to numerical disturbance and correspondingly improves the effective parameters of the model, which results in less prunable parameters. The effective of the proposed PAE-based data augmentation method is proven by the fact that the macroaverage accuracy of the model has exceeded 95%. In the proposed method, there are very few parameters to tune, but still results in higher performance in different model level optimization conditions. Particularly, the proposed PAE-based data augmentation method has no conflicts with the existing model level optimization according to the experimental results. With the fast implementation of the PAE algorithm, the lowcomplexity model can easily achieve over 96% accuracy in the 3-class classification task.

IV. CONCLUSIONS

In this paper, the PAE algorithm is proposed to be used as the data augmentation method for the development of the low-complexity ASC system. A fast implementation of the PAE algorithm is proposed, which is suitable in schemes of STFT and CQT. The usage of the PAE algorithm ensures the correlation between the augmented samples and the original stereo audio sample. Therefore, the proposed method can increase the generalization of data without modifying the original data distribution and be compatible with the existing model level optimization. The effectiveness of the proposed method is validated through experiments, achieving more than 95% macro-average accuracy in the 3-class ASC task.

REFERENCES

- A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop, 2018.
- [2] V. Peltonen, A. Eronen, M. Parviainen, and A. Klapuri, "Recognition of everyday auditory scenes: potential, latencies and cues," in Proceedings of the Audio Engineering Society Convention, 2001.
- [3] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop, 2020.
- [4] Y. Le Cun and Y. Bengio, "Convolutional networks for images, speech, and time series," in The Handbook of Brain Theory and Neural Networks, M. A. Arbib, Ed. Cambridge, MA: MIT Press, 1995, pp. 255–258.
- [5] S. Suh, S. Park, Y. Jeong, and T. Lee, "Designing acoustic scene classification models with CNN variants," in the 2020 IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events, 2020.
- [6] Y. Wu and T. Lee, "Searching for efficient network architectures for acoustic scene classification," in the 2020 IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events, 2020.
- [7] H. Hu, C.-H. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S. M. Siniscalchi, Y. Wang, J. Du, and C.-H. Lee, "Device-robust acoustic scene classification based on two-stage categorization and data augmentation," in the 2020 IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events, 2020.
- [8] M. D. McDonnell, H. Mostafa, R. Wang, and A. Schaik, "Single-bit-perweight deep convolutional neural networks without batch-normalization layers for embedded systems," in 2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), 2019, pp. 197–204.
- [9] S. Chang, J. Cho, H. Park, H. Park, S. Yun, and K. Hwang, "QTI submission to DCASE 2020: model efficient acoustic scene," in the 2020 IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events, 2020.
- [10] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, "CP-JKU submissions to DCASE20: low-complexity cross-device acoustic scene classification with rf-regularized CNNs," in the 2020 IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events, 2020.
- [11] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned CP-Decomposition," in Proceedings of the International Conference on Learning Representations (ICLR), 2015.
- [12] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, "Integrating the data augmentation scheme with various classifiers for acoustic scene modeling," in the 2019 IEEE AASP Detection and Classification of Acoustic Scenes and Events, 2019.
- [13] S. Mun, S. Park, D. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane," in the 2017 IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events, 2017.

- [14] M. Goodwin and J. M. Jot. "Primary-ambient signal decomposition and vector-based localization for spatial audio coding and enhancement," in the 2007 International Conference on Acoustic, Speech, and Signal Processing (ICASSP), 2007.
- [15] H. Yang, C. Shi, and H. Li "Acoustic scene classification using CNN ensembles and primary ambient extraction," in the 2019 IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events, 2019.
- [16] J. He, Spatial audio reproduction with primary ambient extraction, Singapore: Springer Publishing Company, 2016.
- [17] L. Chen, C. Shi, and H. Li, "Primary ambient extraction for random sign Hilbert filtering decorrelation," *Proceedings of the 23rd International Congress on Acoustics*, Aachen, Germany, 2019.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [19] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: beyond empirical risk minimization," in International Conference on Learning Representations (ICLR), 2018.