MULTI-SCALE ARCHITECTURE AND DEVICE-AWARE DATA-RANDOM-DROP BASED FINE-TUNING METHOD FOR ACOUSTIC SCENE CLASSIFICATION

Joo-Hyun Lee, Jeong-Hwan Choi*, Pil Moo Byun*, and Joon-Hyuk Chang*

Department of Electronic Engineering, Hanyang University, Seoul, Republic of Korea
{jhyun42, brent1104, fordream0309, jchang}@hanyang.ac.kr

ABSTRACT

We propose a low-complexity acoustic scene classification (ASC) model structure suitable for short-segmented audio and fine-tuning methods for generalization to multiple recording devices. Based on the state-of-the-art architecture of the ASC, broadcasting-ResNet (BC-ResNet), we introduce BC-Res2Net that uses hierarchical residual-like connections within the frequency- and temporal-wise convolutions to extract multiscale features while using fewer parameters. We also incorporate the attention and aggregation method proposed in short-utterance speaker verification with BC-Res2Net to achieve high performance. In addition, we train the model with a novel fine-tuning method using a device-aware data-random-drop to avoid optimization for only a few devices. When the amount of data differed for each device in the training dataset, the proposed method gradually dropped the data of the primary device from the mini-batch. The experimental results on the TAU Urban Acoustic Scenes 2022 Mobile development dataset demonstrated the effectiveness of multi-scale modeling in short audio. Furthermore, the proposed training strategy significantly reduced the multi-class cross-entropy loss for various devices.

Index Terms— Acoustic scene classification, multi-scale, data imbalance, fine-tuning, short-segmented audio

1. INTRODUCTION

Remarkable progress in acoustic scene classification (ASC) has been accomplished with the development of deep learning, and several studies have recently been conducted to implant deep neural networks (DNNs) into low-resource devices that are suitable for practical applications [1, 2]. Notably, the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge have been held with various audio tasks, including ASC, and contributes to advance computational environmental audio analysis techniques. In DCASE Task 1, ASC systems should satisfy constraints such as data imbalance depending on the recording device, low complexity limitations, and short audio input while ensuring high classification accuracy [3, 4].

Owing to the limitation on the number of parameters, convolutional neural networks (CNNs) are preferred over DNNs for ASC models. To further enhance the feature extraction capability of CNNs, the ResNet [5] structure and depth-wise separable CNNs (DW-CNNs) [6] were adopted in [7] and [8], respectively. Moreover, the modified MobileNet [9], EfficientNet [10], and broadcasting-ResNet (BC-ResNet) [11] structures, which were designed to consider computational power, exhibited excellent performance [3, 12]. To improve the generalization of the model to the multiple devices, ResNorm was proposed with BC-ResNet, which normalized the frequency bands with the residual path [12].

Several studies introduced the model architecture in the speaker verification field, which focused on the frequency of speech and channels in CNN to improve the verification performance in short-duration speech [13–15]. Liu *et al*. [14] conducted a multiscale frequency-channel attention (MFA) framework with frequency-channel attention and Res2Net structure [16], which learned multiscale features to emphasize the significant frequency and channel components. Jung *et al*. [15] proposed a feature pyramid module (FPM) that upsamples in a top-down pathway to effectively aggregate various-resolution feature maps.

This paper introduces two strategies for generalized ASC for various devices under low complexity and short input time conditions: improving the model structure, and training with a novel fine-tuning method. Inspired by Res2Net, we propose BC-Res2Net that are accomplished by modifying the BC-ResNet structure with multiscale modeling to increase the receptive field size of each CNN. We integrate the BC-Res2Net with MFA and FPM, which effectively extract and aggregate features from short speech signals, to construct the ASC model. Subsequently, we perform device-aware data-random-drop-based fine-tuning that drops data of the selected device in batch-level processing for the pretrained model to obtain consistent performance in various recording devices. We choose the device that recorded the most in the training dataset. We do not drop the data at the beginning of the fine-tuning but gradually increased the drop rate to a given parameter. In addition, we add regularization with a cross-entropy loss to avoid overfitting devices that are not selected.

2. ASC MODEL ARCHITECUTRE

2.1. BC-Res2Net structure

Broadcasted residual learning [11] residually connects two-dimensional (2D) and one-dimensional (1D) feature maps with the input. These different sized feature maps are extracted by frequency-wise 2D and temporal-wise 1D DW-CNNs, respectively, and they contain frequency and frequency-aware temporal features. To correct the size mismatch, the output of the 1D DW-CNN, a 1D feature map is broadcast along the frequency axis. This residual connection was combined with basic structures such as ResNet and Transformer [17] to obtain state-of-the-art results in various audio and speech fields [11, 12, 18] where it is paramount to effectively capture frequency-time characteristics.

Res2Net, however, computes more efficiently than the conventional convolution-based structure because it comprises several CNNs connected in a hierarchical residual-like manner. In particular, the input feature map is sliced precisely with the same channel

*Equal contributions.

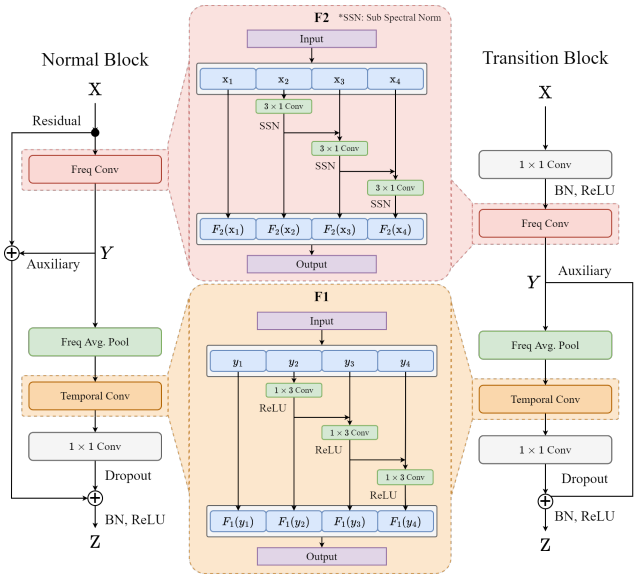

Figure 1: BC-Res2Net structure.

size of CNNs. Each partitioned feature subset has a corresponding CNN that is fed while being sequentially added to the output of the previous CNN. The CNN outputs have various receptive field sizes owing to the multiscale operation. To extract the frequency and frequency-aware temporal features in a multiscale manner, we propose the BC-Res2Net that converts the frequency- and temporal-wise convolution of the BC-ResNet into a Res2Net structure. Figure 1 shows the network blocks of the BC-Res2Net comprising $\mathbf{F}_2$, $\mathbf{F}_1$, frequency average pooling, and point-wise 1D CNN, where $\mathbf{F}_2$ and $\mathbf{F}_1$ denote the Res2Net style 2D and 1D convolutions including nonlinear functions, respectively. Because the transition block is used when the given input and output feature map sizes are assigned differently, we describe the proposed structure based on the normal blocks.

The input feature map $\mathbf{X} \in \mathbb{R}^{C\times F\times T}$ is sliced into S subfeature maps along with the channel axis, and the feature map subset is denoted by $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_s\}$. Except for the first subfeature map $\mathbf{x}_1$, each $\mathbf{x}_i \in \mathbb{R}^{C/S\times F\times T}$ has a corresponding frequency-wise 2D DW-CNN with a kernel size of 3×1 and subspectral normalization [19], denoted by $f_{2C,i}(\cdot)$ and $f_{SSN,i}(\cdot)$, respectively. The feature subset $\mathbf{x}_i$ is sequentially fed into $f_{2C,i}(\cdot)$ and $f_{SSN,i}(\cdot)$, after adding the output of $f_{SSN,i-1}(\cdot)$. The overall process for $\mathbf{F}_2$ can be formulated as follows:

$$\mathbf{F}_2(\mathbf{x}_i) = \begin{cases} \mathbf{x}_i, & \text{if } i = 1 \\ f_{SSN,i}(f_{2C,i}(\mathbf{x}_i)), & \text{if } i = 2 \\ f_{SSN,i}(f_{2C,i}(\mathbf{x}_i + \mathbf{F}_2(\mathbf{x}_{i-1}))), & \text{otherwise} \end{cases} \quad (1)$$

Frequency average pooling is applied after concatenating the set of outputs $\{\mathbf{F}_2(\mathbf{x}_1), \mathbf{F}_2(\mathbf{x}_2), ..., \mathbf{F}_2(\mathbf{x}_s)\}$ along the channel axis. The obtained feature map $Y \in \mathbb{R}^{C\times 1\times T}$ is then fed into $\mathbf{F}_1$ to extract the temporal characteristics. $\mathbf{F}_1$ can be expressed as follows:

$$\mathbf{F}_1(y_i) = \begin{cases} y_i, & \text{if } i = 1 \\ f_{ReLU,i}(f_{1C,i}(y_i)), & \text{if } i = 2 \\ f_{ReLU,i}(f_{1C,i}(y_i + \mathbf{F}_1(y_{i-1}))), & \text{otherwise} \end{cases} \quad (2)$$

where $y_i \in \mathbb{R}^{C/S\times 1\times T}$ denotes the subfeature map of Y that is sliced into S along with the channel axis, and $f_{1C,i}$, and $f_{ReLU,i}(\cdot)$ denote a corresponding temporal-wise 1D DW-CNN with a kernel size of 3 and the ReLU activation, respectively. The following operations are performed sequentially: concatenating outputs of $\mathbf{F}_1$ into one, point-wise convolution, channel-wise dropout, and expanding the feature map size $\mathbb{R}^{C\times 1\times T}$ to $\mathbb{R}^{C\times F\times T}$ along with the frequency axis. Finally, batch normalization with ReLU activation is applied after combining the output with two auxiliary residuals that the input identity and result of $\mathbf{F}_1$. Note that the BC-Res2Net operates with fewer CNN parameters than the BC-ResNet because the first subfeature map sliced from $\mathbf{F}_2$ and $\mathbf{F}_1$ does not proceed with convolution. The transition block differs from the normal block in two ways: auxiliary point-wise 2D CNN is applied before $\mathbf{F}_2$ to change the input channel size, and there is no residual connection for identity due to the size difference between the input and output.

2.2. ASC model for MFA and FPM

Short audio or speech makes feature extraction difficult because of insufficient temporal information. Several studies have focused on adding or enhancing the DNN structure in speaker verification fields to overcome performance degradation under short-utterance situations. In this study, we combined MFA and FPM, which improved the feature maps using the attention mechanism and aggregated the features from multiple resolutions, respectively, with the BC-Res2Net structure to introduce the ASC model for short audio.

Table 1: Architectures of proposed BC-Res2Net-based ASC model. T ,F, and C denote the number of time sequences, frequency bins, and CNN channel respectively. Input feature size is $1 \times F \times T$.

Output size	Stage	Operator
$2C \times F/2 \times T/2$	Stem	Conv2D $[5 \times 5]$, stride 2 BatchNorm + MFA
$C \times F/2 \times T/2$	Stage 1	BC-Res2Net $\times$ 2 ResNorm + MFA
$1.5C \times F/4 \times T/4$	Stage 2	Max-pool $[2 \times 2]$ BC-Res2Net $\times$ 2 ResNorm + MFA
$2C \times F/8 \times T/8$	Stage 3	Max-pool $[2 \times 2]$ BC-Res2Net $\times$ 2 ResNorm + MFA
$2.5C \times F/8 \times T/8$	Stage 4	BC-Res2Net $\times$ 3 ResNorm + MFA
$4C \times 1 \times 1$	Aggregation	FPM
# Classes $\times 1 \times 1$	Classifier	Linear

The overall architecture is presented in Table 1. Assigning the importance of the frequency channel components for all the outputs is necessary because the output of each stage has a different resolution and receptive field. Therefore, we apply MFA to Stem and every stage after ResNorm. Figure 2 shows the FPM aggregating the MFA output of each stage in a bottom-up pathway, where the base CNN channel of the proposed model is set to 40. The last three feature maps are upsampled to the size of $\mathbb{R}^{C\times F/2\times T/2}$, which is the same as the output of Stage 1. We considered the average of four feature maps across the frequency-temporal dimension and concatenated them into a single feature map with a size of $\mathbb{R}^{4C\times 1\times 1}$. The upsampling method of the FPM is converted into the pixel shuffle method [20] for greater efficiency compared to the transposed convolution method proposed in [15]. The output of FPM is reshaped through a linear layer according to the number of classes.

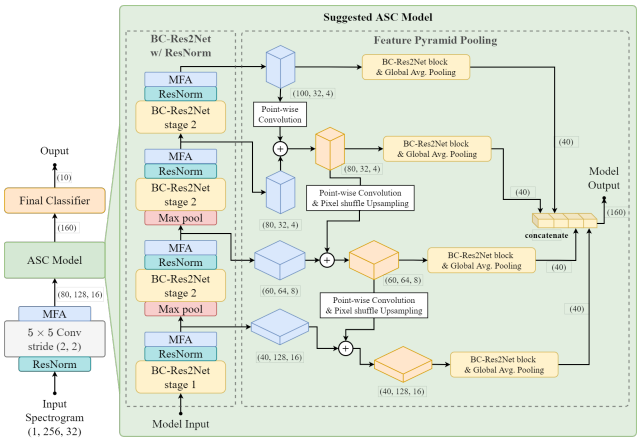

Figure 2: Proposed BC-Res2Net-based ASC model architecture.

3. FINE-TUNING METHOD USING DEVICE-AWARE DATA-RANDOM-DROP

When we train the ASC model with a significantly different amount of data for each device, the model undertakes greater effort to characterize a few specific devices occupying a large percentage of the dataset than to represent the general features of the overall device. To alleviate this issue, we load a pre-learning model that neglected the device types and fine-tuned the model by removing data from specific devices. Before fine-tuning, the last linear layer that classified the scenes is initialized and retrained to become more device-agnostic. Next, we fine-tune the model using the suggested device-aware data-random-drop method. It selects the device with the most data from the dataset and manages the mini-batch for every step by randomly removing data from it. However, an excessive drop of specific devices from the training process can lead to poor ASC performance owing to a lack of data diversity Therefore, inspired by curriculum learning [21], we design a method in which the data-drop rate is initialized with zero and then increased step-wise. The drop rate gradually increases with the shape of the sigmoid function from zero to the given parameter. Furthermore, we add regularization to minimize the square weight difference between the fine-tuning and pretrained model parameters to prevent excessive loss of information from the selected device as given by:

$$L = L_C + \lambda \sum_i (\theta_i - \widetilde{\theta}_{P,i})^2, \tag{3}$$

where L, L_C, and λ denote the total loss, classification loss, and scaling factor of regularization, respectively. θ and $\widetilde{\theta}_P$ denote the fine-tuned and pretrained model parameters, respectively, except for the classifier layer. The pretrained model parameters are stored in advance.

4. EXPERIMENTAL SETUP

4.1. Datasets and preprocessing

The TAU Urban Acoustic Scenes 2022 Mobile Development dataset [4] had the same format as the 2020 development dataset [3], same sample rate of 44.1 kHz and 24 bits. However, this 2022 dataset segments were significantly shorter (1 s) compared to the last 2020 development set (10 s). In addition, the number of segments grew tenfold as the 2020 dataset split the 2022 dataset by 1 s.

Table 2: Ablation study of the BC-Res2Net evaluated on the TAU Urban Acoustic Scenes 2022 Mobile development dataset. (Acc. indicates the top-1 test accuracy(%).)

Systems	# Params	MACs	Log Loss	Acc.
BC-ResNet-40	88.1K	17.21M	1.327	57.1
BC-Res2Net-40	85.8K	15.89M	1.235	59.1
w/ MFA	123.6K	17.45M	1.198	59.3
w/ FPM	93.6K	17.06M	1.212	59.5
w/ MFA & FPM	126.6K	26.76M	**1.167**	**60.8**

Table 3: Log loss and top-1 test accuracy (%) comparison for different duration of test audio on the TAU Urban Acoustic Scenes 2020 Mobile development dataset. (Dur. indicates durations.)

Dur.	BC-ResNet-40	BC-Res2Net-40	BC-Res2Net-40 w/ MFA & FPM
1 s	1.327 / 57.1	1.235 / 59.1	**1.167 / 60.8**
2 s	1.285 / 57.8	1.190 / 60.3	**1.146 / 61.6**
5 s	1.301 / 56.7	1.185 / 59.7	**1.172 / 60.5**
10 s	1.315 / 56.3	1.195 /58.7	**1.192 / 59.5**

The audio segments were ten types of acoustic scenes from ten cities, recorded from three real devices (A, B, and C) and six simulated devices (S1–S6). According to the train-split method of [4], development dataset 2022 was separated into training and test subsets comprising 139,970 and 29,680 segments, respectively. In the training subset, the data for Device A accounted for 73% of the total. In the test split, the data from all the devices were evenly distributed. The test split contained data recorded with devices S4-S6, which were excluded in the training data split. The evaluation dataset was provided without labels for submitting the results.

We used the log Mel spectrum as the input feature for our system. The input features were prepared through three steps: downsampling from 44.1 kHz to 16.0 kHz, log Mel spectrum feature extraction, and data augmentation. The log Mel spectrograms were 256-dimensional, extracted with 2048 samples of the Hanning window, and 512 sample shifts. The input feature size obtained using the preprocessing method mentioned was [1, 256, 32]. The time-rolling method was used for time-domain augmentation. The input audio was randomly rolled along the time axis, ranging from -0.5-0.5 s, with out-of-range parts shifted to the other side. Specaugment [22], except time wrapping, was also employed with two frequency and temporal masks each. Mask parameters of 40 and 4 were used for the frequency and temporal masks, respectively. Each time-rolling and Specaugment mask was applied with a probability of 0.8. We also applied Mixup [23] with $\alpha = 0.3$ to the acoustic feature space.

4.2. Implementation details

We trained the BC-Res2Net-based ASC model with pretraining and fine-tuning phases. In the pretraining phase, the AdamW optimizer [24] with a weight decay of 0.05 was used over 300 epochs, and the mini-batch size was set to 512. Warmup [25] was applied, where the learning rate linearly increased from 1e-8 to 0.01 over the first ten epochs and decayed to zero with a cosine annealing scheduler [26]. We applied a device-aware data-random-drop, treating the selected Device A as an excluded recording device in the fine-tuning phase. The mixup was disabled to correct the mismatch between the training and test conditions. The scaling factor of regularization was 0.4, and the AdamW optimizer with a weight decay of 1e-8 and fixed

Table 4: Device-wise top-1 test accuracy (%) and overall log loss comparison of proposed fine-tuning method according to maximum drop rate on the TAU Urban Acoustic Scenes 2022 Mobile development dataset. (Acc. indicates the top-1 test accuracy (%).)

Systems	Fine-tuning (Drop rate)	Seen device						Unseen device			Average Log loss / Acc.
		A	B	C	S1	S2	S3	S4	S5	S6	
BC-Res2Net-40 w/ MFA & FPM	✗	72.0	64.5	68.2	62.7	59.5	64.2	54.4	56.4	45.3	1.167 / 60.8
	✓ (0.00)	72.8	**68.1**	69.7	63.1	62.0	66.2	53.4	56.0	47.6	1.083 / 62.1
	✓ (0.50)	73.0	67.8	70.1	62.9	61.6	**66.5**	55.2	56.8	48.4	1.085 / 62.5
	✓ (0.90)	**73.2**	68.0	69.4	**63.7**	61.6	66.2	55.2	57.3	**49.0**	**1.076 / 62.6**
	✓ (0.99)	72.7	67.2	**70.3**	63.0	**62.2**	66.0	**55.9**	**57.4**	48.8	1.081 / 62.6

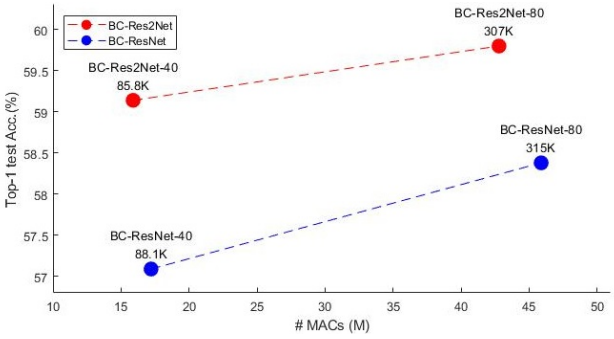

Figure 3: Top-1 test accuracy comparison of the BC-Res2Net and the BC-ResNet according to model parameters and MACs.

learning rate of 1e-5 was used. For the model structure, both $\mathbf{F}_2$ and $\mathbf{F}_1$ of the BC-Res2Net were sliced into four subchannels. Sub-spectral normalization [19] with four sub-bands and ResNorm with 0.1 (hyperparameter of the identity shortcut path,) were applied to the BC-Res2Net.

5. RESULT

We evaluated the result in terms of the top-1 test accuracy and multiclass cross-entropy (log loss). We also reported the number of model parameters and multiply-accumulate operations (MACs) were used to observe computational complexity. Figure 3 shows the comparison of the BC-ResNet and BC-Res2Net when the base CNN channel size increases from 40 to 80. For all the CNN channel sizes, the BC-Res2Net achieves higher accuracy than the BC-ResNet while having small MACs and model parameters. Table 2 presents the effects of the proposed structural modifications on the ASC model. The BC-Res2Net-40 requires 2.6% fewer parameters and 7.7% fewer MACs than the BC-ResNet-40 but performs better in terms of log loss and accuracy. When MFA and FPM were applied to the BC-Res2Net, the accuracy improved by 0.2% and 0.3%, respectively; when both were applied, the accuracy improved by 1.7%. Table 3 shows the results for the short audio conditions. We evaluated the cropped test data with the given duration within each 10 s audio of the 2020 data. The BC-Res2Net performed better than the BC-ResNet for all the test lengths; in particular, the model that added MFA and FPM to the BC-Res2Net obtained better results at shorter durations of 1 s and 2 s. These results show that the BC-Res2Net extracts the information required to classify the scenes more effectively than the BC-ResNet, and using MFA and FPM additionally assists in classifying sound in short-segmented audio. Table 4 presents the effect of the fine-tuning method based on the maximum drop rate. Compared with the pretrained model, the overall accuracy and log loss improved by 0.084 and 1.3%, respectively; when the fine-tuning was applied without data-random-drop, and better performance was achieved when the drop was applied to the selected Device A. Maximum drop rate of 0.9 exhibited the best average log loss and accuracy, and achieved significant improvements of 0.091 and 1.8%, respectively, compared to the case when fine-tuning was not applied. In particular, on the seen device, the performance of multiple devices including Device A was improved evenly, and the performance improvement was observed even in the unseen device, showing that the proposed fine-tuning method benefits generalization of the device.

6. RELATIONSHIPS WITH TECHNICAL REPORT

In a technical report [27], quantization-aware training (QAT) [28] was additionally introduced to satisfy the quantization conditions of INT8. The log loss and accuracy of the QAT-applied BC-Res2Net-40-based ASC model were degraded to be 1.193 and 60.3%, respectively, compared with the results without quantization. We submitted the outputs of the two systems with the proposed fine-tuning in QAT environment. To investigate the effect of the proposed fine-tuning according to the regularization scaling factor, we submitted systems results trained with the fine-tuning method with a drop rate of 0.9 and the regularization factors with 0.04 and 0.4. Each result achieved log losses of 1.072 (Acc. 62.2%) and 1.065 (Acc. 62.6%), respectively, for the test set of the development dataset. For the rest of the two trials, we additionally applied knowledge distillation [29] introduced in [7] to improve the performance. The best result was assigned the teacher model size equal to the student model and the scaling factor of regularization to 0.4 and achieved a log loss of 0.835 and accuracy of 70.1% from the development dataset. Finally, this result achieved a log loss of 1.147 and accuracy of 60.8% in the challenge evaluation and placed second in the competition.

7. CONCLUSION

We proposed the BC-Res2Net by modifying the BC-ResNet in a multiscale manner. Moreover, we improved ASC performance under short audio evaluation conditions by using the MFA and the FPM method, which finds important components among frequency and channel components and effectively aggregates feature maps of different resolutions. Finally, we suggested the device-aware data-random-drop method-based fine-tuning method to promote optimization for multiple devices.

8. ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00456, Development of Ultra-high Speech Quality Technology for Remote Multi-speaker Conference System)

9. REFERENCES

[1] M. Valenti *et al*., "DCASE 2016 acoustic scene classification using convolutional neural networks," in *Proc. Workshop Detection Classif. Acoust. Scenes Events*, 2016, p. 95–99.

[2] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time–frequency audio features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[3] I. Martín-Morató, A. Ancilotto, T. Heittola, A. Mesaros, and T. Virtanen, "Low-complexity acoustic scene classification for multi-device audio: analysis of DCASE 2021 challenge systems," *arXiv:2105.13734*, 2021.

[4] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in DCASE 2022 challenge," *arXiv:2206.03835*, 2022.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv:1512.03385*, 2015.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861*, 2017.

[7] B. Lehner and K. Koutini, "Acoustic scene classification with reject option based on ResNets," *DCASE 2019 Challenge, Tech. Rep.*, 2019.

[8] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: generalization across devices and low complexity solutions," in *Proc. Detection and Classification of Acoustic Scenes and Events* (DCASE) *Workshop*, 2020, p. 56–60.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2018, p. 4510–4520.

[10] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. International Conference on Machine Learning* (ICML), vol. 97, 2019, pp. 6105–6114.

[11] B. Kim, S. Yang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," in *Proc. INTERSPEECH*, 2021, p. 4538–4542.

[12] B. Kim, S. Yang, J. Ki, and S. Chang, "Domain generalization on efficient acoustic scene classification using residual normalization," in *Proc. Detection and Classification of Acoustic Scenes and Events* (DCASE) *Workshop*, 2021, p. 21–25.

[13] J.-H. Choi, J.-Y. Yang, and J.-H. Chang, "Short-utterance embedding enhancement method based on time series forecasting technique for text-independent speaker verification," in *IEEE Automatic Speech Recognition and Understanding Workshop* (ASRU), 2021, pp. 130–137.

[14] T. Liu, R. K. Das, K. Aik Lee, and H. Li, "MFA: TDNN with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2022, pp. 7517–7521.

[15] Y. Jung, S. M. Kye, Y. Choi, M. Jung, and H. Kim, "Improving multi-scale aggregation using feature pyramid module for robust speaker verification of variable-duration utterances," in *Proc. INTERSPEECH*, 2020, pp. 1501–1505.

[16] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, p. 652–662, Feb. 2021.

[17] A. Vaswani *et al*., "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems* (NIPS), 2017, p. 3830–3834.

[18] J.-H. Choi, J.-Y. Yang, Y.-R. Jeoung, and J.-H. Chang, "Improved CNN-Transformer using broadcasted residual learning for text-independent speaker verification," in *Proc. INTERSPEECH*, 2022.

[19] S. Chang, H. Park, J. Cho, H. Park, S. Yun, and K. Hwang, "Subspectral normalization for neural audio data processing," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 2021, pp. 850–854.

[20] W. Shi *et al*., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2016, pp. 1874–1883.

[21] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. International Conference on Machine Learning* (ICML), 2009, p. 41–48.

[22] D. S. Park *et al*., "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. INTERSPEECH*, 2019, pp. 2613–2618.

[23] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. International Conference on Learning Representations* (ICLR), 2018.

[24] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. International Conference on Learning Representations* (ICLR), 2019.

[25] P. Goyal, "Accurate, large minibatch SGD: Training imagenet in 1 hour," *arxiv:1706.02677*, 2017.

[26] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. International Conference on Learning Representations* (ICLR), 2017.

[27] J.-H. Lee, J.-H. Choi, P. M. Byun, and J.-H. Chang, "HYU submission for the DCASE 2022: fine-tuning method using device-aware data-random-drop for device-imbalanced acoustic scene classification," *DCASE 2022 Challenge, Tech. Rep.*, 2022.

[28] https://pytorch-lightning.readthedocs.io/en/stable/_modules/pytorch_lightning/callbacks/quantization.html.

[29] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learning and Representation Learning Workshop*, 2015.