

QTI SUBMISSION TO DCASE 2021: RESIDUAL NORMALIZATION FOR DEVICE-IMBALANCED ACOUSTIC SCENE CLASSIFICATION WITH EFFICIENT DESIGN

Technical Report

Byeonggeun Kim¹, Seunghan Yang¹, Jangho Kim^{1,2,*}, Simyung Chang¹

¹Qualcomm AI Research[†], Qualcomm Korea YH, Seoul, Republic of Korea

²Seoul National University, Seoul, Republic of Korea

{kbungkun, seunghan, jangkim, simychan}@qti.qualcomm.com

ABSTRACT

This technical report describes the details of our TASK1A submission of the DCASE2021 challenge. The goal of the task is to design an audio scene classification system for device-imbalanced datasets under the constraints of model complexity. This report introduces four methods to achieve the goal. First, we propose Residual Normalization, a novel feature normalization method that uses instance normalization with a shortcut path to discard unnecessary device-specific information without losing useful information for classification. Second, we design an efficient architecture, BC-ResNet-Mod, a modified version of the baseline architecture with a limited receptive field. Third, we exploit spectrogram-to-spectrogram translation from one to multiple devices to augment training data. Finally, we utilize three model compression schemes: pruning, quantization, and knowledge distillation to reduce model complexity. The proposed system achieves an average test accuracy of 76.3% in TAU Urban Acoustic Scenes 2020 Mobile, development dataset with 315k parameters, and average test accuracy of 75.3% after compression to 61.0KB of non-zero parameters.

Index Terms— acoustic scene classification, efficient neural network, domain imbalance, residual normalization, model compression

1. INTRODUCTION

Acoustic scene classification (ASC) is the task of classifying sound scenes such as “airport”, “train station”, and “urban park” to which a user belongs. ASC is an important research field that plays a key role in various applications such as context-awareness and surveillance [1, 2, 3]. Detection and Classification of Acoustic Scenes and Events (DCASE) [4] is an annual challenge, attracting attention to the field and encouraging research. There are various interesting tasks in the DCASE2021 challenge, and we aim for TASK1A: Low-Complexity Acoustic Scene Classification with Multiple Devices [5, 6].

TASK1A classifies ten different audio scenes from 12 European cities using four different devices and 11 additional simulated devices. This year, the task becomes more challenging as an ASC model needs to meet two goals simultaneously; First, data is collected from multiple devices, and the number of samples per device is unbalanced. Therefore, the proposed system needs to solve the

domain imbalance problem while generalizing to different devices. Second, TASK1A restricts the model size and therefore requires an efficient network design.

In recent years, a number of researches have been proposed for more efficient and high-performance ASC. Most of them are based on convolutional neural network (CNN) using residual network and ensemble [7, 8, 9, 10]. The top-performing models in the previous TASK1A utilize multiple CNNs in a single model with parallel connections [7, 9]. For the generalization of the model, [8, 11] show that there is a regularization effect by adjusting the receptive field size in CNN-based design. However, these works also use models of several MB, and it is still challenging to satisfy the low model complexity of TASK1A of this year. In addition, when using the previous methods, we found an accuracy drop of up to 20% on the unseen devices compared to the device with sufficient training data. In this work, we propose methods to leverage the generalization capabilities of unseen devices while maintaining the model’s performance in lightweight models. First, we introduce a network architecture for ASC that utilizes broadcasted residual learning [12]. Based on this architecture, we can achieve higher accuracy while reducing the size by a third of the baseline [8]. Next, we propose a novel normalization method, Residual Normalization (ResNorm), which can leverage the generalization performance for unseen devices. ResNorm allows maintaining classification accuracy while minimizing the influence on different frequency responses of devices by performing normalization of frequency bands in the residual path. Furthermore, we introduce a device translator that converts data from one to other devices. With the aid of data augmentation with the translator, we can mitigate the domain imbalance problem more as well as the domain difference among multiple devices. Finally, we describe model compression combined with pruning and quantization to satisfy the model complexity of the task while maintaining performance using knowledge distillation.

The rest of the report is organized as follows. Section 2 describes the network architecture, Residual Normalization, device translator, and model compression methods. Section 3 shows the experimental results and analysis. Finally, we conclude the work in Section 4.

2. PROPOSED METHOD

This session introduces an efficient model design for the task of device-imbalanced acoustic scene classification. First, we present a modified version of Broadcasting residual network [12] for the acoustic scene domain. Following, we propose Residual Normal-

[†] Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.* Author completed the research in part during an internship at Qualcomm Technologies, Inc.

Table 1: **BC-ResNet-Mod**. Each row is a sequence of one or more identical modules repeated n times with input shape of frequency by time by channel and total time step T .

Input	Operator	n	Channels
$256 \times T \times 1$	conv2d 5x5, stride 2	-	2c
$128 \times T/2 \times 2c$	stage1: BC-ResBlock	2	c
$128 \times T/2 \times c$	max-pool 2x2	-	-
$64 \times T/4 \times c$	stage2: BC-ResBlock	2	1.5c
$64 \times T/4 \times 1.5c$	max-pool 2x2	-	-
$32 \times T/8 \times 1.5c$	stage3: BC-ResBlock	2	2c
$32 \times T/8 \times 2c$	stage4: BC-ResBlock	3	2.5c
$32 \times T/8 \times 2.5c$	conv2d 1x1	-	num class
$32 \times T/8 \times \text{num class}$	avgpool	-	-
$1 \times 1 \times \text{num class}$	-	-	-

ization for generalization in a device-imbalanced dataset. Next, we explain a spectrogram-to-spectrogram translation network to convert data from one device to other devices. Finally, we describe how to get a compressed version of the proposed system.

2.1. Network Architecture

To design a low-complexity network in terms of the number of parameters, we use a Broadcasting-residual network (BC-ResNet) [12], a baseline architecture that uses 1D and 2D CNN features together for better efficiency. While the [12] targets human voice, we aim to classify the audio scenes. Therefore, we make two modifications to the network, *i.e.*, limit the receptive field and use max-pool instead of dilation, to adapt to the differences in input domains.

The proposed architecture is shown in Table 1, a fully CNN named modified BC-ResNet (BC-ResNet-Mod). The model has 5x5 convolution on the front with a 2x2 stride for downsampling followed by BC-ResBlocks [12]. In [8], they show that the size of the receptive field can regularize CNN-based ASC models. We change the depth of the network and use max-pool to control the size of the receptive field. With a total of 9 BC-ResBlocks and two max-pool layers, the receptive field size is 109x109. We also do the last 1x1 convolution before global average pooling that the model classifies each receptive field separately and ensembles them by averaging. BC-ResNets use dilation in temporal dimension to obtain a larger receptive field while maintaining temporal resolution across the network. We observe that time resolution does not need to be fully kept in the audio scene domain, and instead of dilation, we insert max-pool layers in the middle of the network.

2.2. Residual Normalization

Instance normalization (IN) [13] is a representative approach to reducing unnecessary domain gaps for better domain generalization [14] or domain style transfer [15] in the image domain. While domain difference can be captured by channel mean and variance in the image domain, we observe that differences between audio devices are revealed along frequency dimension. To get audio device generalized features, we use instance normalization by frequency (FreqIN) as below.

$$\text{FreqIN}(x) = \frac{x - \mu_{nf}}{\sqrt{\sigma_{nf}^2 + \epsilon}}, \quad (1)$$

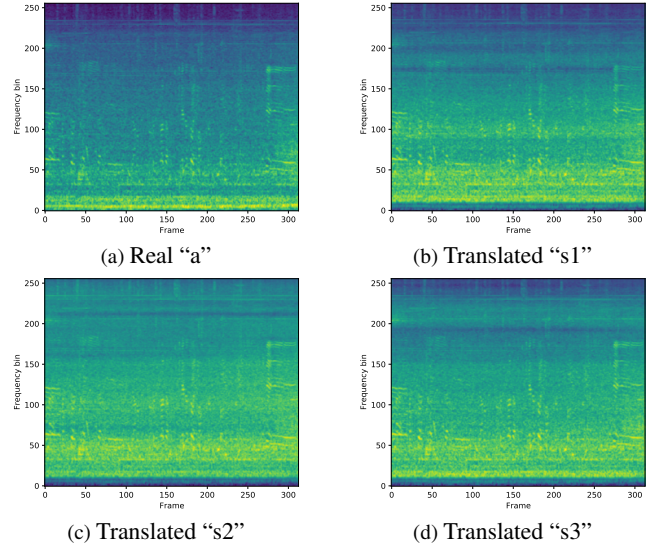


Figure 1: **Examples of device translation**. (b), (c), and (d) are generated by the device translator using the real data (a).

where,

$$\begin{aligned} \mu_{nf} &= \frac{1}{CT} \sum_{c=1}^C \sum_{t=1}^T x_{nct}, \\ \sigma_{nf}^2 &= \frac{1}{CT} \sum_{c=1}^C \sum_{t=1}^T (x_{nct} - \mu_{nf})^2. \end{aligned} \quad (2)$$

Here, $\mu_{nf}, \sigma_{nf} \in \mathbb{R}^{N \times F}$ are mean and standard deviation of the input feature $x \in \mathbb{R}^{N \times C \times F \times T}$, where N, C, F, T denote batch size, number of channel, frequency dimension, and time dimension respectively. ϵ is a small number added to avoid division by zero.

Direct use of IN can result in loss of useful information for classification contained in domain information. To compensate for information loss due to FreqIN, we add an identity path multiplied by a hyperparameter λ . We suggest a normalization method, named Residual Normalization (ResNorm) which is

$$\text{ResNorm}(x) = \lambda \cdot x + \text{FreqIN}(x). \quad (3)$$

We apply ResNorm for input features and after the end of every stage in Table 1. There are a total of five ResNorm modules in the network.

2.3. Device Translator

We design the device translator, which converts a spectrogram from one device into another. We start with the pix2pix framework [16], which has been shown to produce high-quality results in image-to-image translation. As in pix2pix, we exploit an encoder-decoder network and a discriminator, but we redesign the encoder-decoder suitable for the audio domain. To consider the characteristics of audio, we adapt Subspectral Normalization [17] to the encoder-decoder network. In addition, we observe that the U-Net with a large depth can reconstruct the spectrogram well. Therefore, we use the U-Net architecture [18] as the encoder-decoder and insert several residual blocks in the middle of the network.

Table 2: **Overall test results.** Top-1 test accuracy (%) and standard deviation on TAU Urban AcousticScenes 2020 Mobile, development dataset for each device (A to S6). Device S4, S5 and S6 remain unseen in training (averaged over 3 seeds, * averaged over 6 seeds).

Method	#Param	A	B	C	S1	S2	S3	S4	S5	S6	Overall
BC-ResNet-Mod-1	8.1k	73.1	61.2	65.3	58.2	57.3	66.2	51.5	51.5	46.3	58.9 ± 0.8
BC-ResNet-Mod-1 + Global FreqNorm	8.1k	73.9	60.9	65.5	60.2	57.9	67.9	50.2	54.3	49.4	60.0 ± 0.9
BC-ResNet-Mod-1 + FreqIN	8.1k	69.9	63.5	60.0	65.3	66.7	67.6	65.9	64.9	62.0	65.1 ± 0.6
BC-ResNet-Mod-1 + Pre-ResNorm	8.1k	75.1	68.9	67.0	66.0	63.9	69.3	63.4	66.9	63.6	67.1 ± 0.8
BC-ResNet-Mod-1 + ResNorm	8.1k	76.4	65.1	68.3	66.0	62.2	69.7	63.0	63.0	58.3	*65.8 ± 0.7
CP-ResNet, c=64	899k	77.0	69.3	69.6	70.3	68.2	70.9	62.7	63.9	58.1	67.8
BC-ResNet-8, num SSN group = 4	317k	77.9	70.4	72.4	69.5	68.3	69.8	66.3	64.1	58.6	68.6 ± 0.4
BC-ResNet-Mod-8	315k	80.7	72.8	74.4	71.4	68.7	71.0	62.2	65.3	59.4	69.5 ± 0.3
BC-ResNet-Mod-8 + Pre-ResNorm	315k	80.8	73.7	73.0	74.0	72.9	77.8	73.3	72.1	71.0	74.3 ± 0.3
BC-ResNet-Mod-8 + ResNorm	315k	81.3	74.4	74.2	75.6	73.1	78.6	73.0	74.0	72.7	* 75.2 ± 0.4
BC-ResNet-Mod-8 + ResNorm, Device Translator	315k	80.5	74.4	73.9	76.0	73.2	78.5	74.1	74.1	73.6	75.4 ± 0.3
BC-ResNet-Mod-8 + ResNorm, 300epoch, KD	315k	82.6	75.6	74.7	77.0	74.2	78.7	75.1	74.8	73.4	*76.3 ± 0.8
+ model compress	-	82.0	73.8	74.3	76.2	73.2	78.8	73.8	72.8	73.3	*75.3 ± 0.8

We train the network with two objectives, a reconstruction loss and a GAN-based loss. We use L2 distance as a reconstruction loss for synthesizing the spectrogram. It helps the network to produce similar output to the known paired ground truth image. In addition, we utilize LSGAN [19] as a GAN-based loss that is designed to encourage the network generating the image to be realistic and diverse. With these two loss functions, the device translator can convert the spectrogram from one to other devices.

2.4. Model Compression

To compress the proposed model, we utilize three model compression schemes: pruning, quantization, and knowledge distillation.

Pruning. The pruning method prunes unimportant weights or channels based on many criteria. In this work, we choose a magnitude-based one-shot unstructured pruning scheme used in [20]. After training, we conduct unstructured pruning on all convolution layers and do additional training to enhance the pruned model’s performance.

Quantization. Quantization is the method to map continuous infinite values to a smaller set of discrete finite values. We quantize all of our models with quantization-aware training (QAT) with symmetric quantization [20]. We combine the pruning and quantization methods. So, we quantize the important weights which are not pruned after the pruning process in the additional training phase. Basically, we utilize the half-precision representation for all weights. For some models, we quantize all convolution layers as an 8-bit, shown in Table 3.

Knowledge Distillation. Knowledge Distillation (KD) trains the lightweight model using a pre-trained teacher network. In general, previous model compression schemes such as pruning and quantization decrease the performance by reducing the model complexity. To enhance the performance of the compressed model, we use a KD loss [21] using the pre-trained model as a teacher network.

3. EXPERIMENTS

3.1. Experimental Setup

Datasets. We evaluate the proposed method on the TAU Urban Acoustic Scenes 2020 Mobile, development dataset [6]. The dataset consists of a total of 23,040 audio segment recordings from 12 European cities in 10 different acoustic scenes using 3 real devices

(A, B, and C) and 6 simulated devices (S1-S6). The 10 acoustic scenes contain “airport”, “shopping mall”, “metro station”, “pedestrian street”, “public square”, “street with traffic”, “park”, and travelling by “tram”, “bus”, and “metro”. Audio segments from B and C are recorded simultaneously with device A, but not perfectly synchronized. Simulated devices S1-S6 generate data using randomly selected audio segments from real device A. Each utterance is 10-sec-long and the sampling rate is 48kHz. [6] divides the dataset into training and test of 13,962 and 2,970 segments, respectively. In the training data, device A has 10,215 samples while B, C, and S1-S3 have 750 samples each, which means the data is device-imbalanced. Devices S4-S6 remain unseen in training. In test data, all devices from A to S6 have 330 segments each. For the device translator, we use paired data between devices existing in the dataset.

Implementation Details. We do downsampling by 16kHz and use input features of 256-dimensional log Mel spectrograms with a window length of 130ms and a frameshift of 30ms. During training, we augment data to get a more generalized model. In the time dimension, we randomly roll each input feature in the range of -1.5 to 1.5 sec, and the out-of-range part is added to the opposite side. We also use Mixup [22] with $\alpha = 0.3$ and SpecAugment [23] with two frequency masks and two temporal masks with mask parameters of 40 and 80, respectively, except time warping. In BC-ResNet-Mod, we use Subspectral Normalization [17] as indicated in [12] with 4 sub-bands and use dropout rate of 0.1. We train the models for 100 epoch using stochastic gradient descent (SGD) optimizer with momentum to 0.9, weight decay to 0.001, mini-batch size to 64, and learning rate linearly increasing from 0 to 0.06 over the first five epochs as a warmup [24] before decaying to zero with cosine annealing [25] for the rest of the training. For the device translator, we use the U-Net architecture and insert six residual blocks in the middle of the network with Subspectral Normalization with 16 sub-bands. We follow the same training configurations used in [16].

Baselines. We compare our method with simple baselines and some prior works: 1) Global FreqNorm, which normalizes data by global mean and variance of each frequency bin; 2) Pre-ResNorm, which uses ResNorm as preprocessing module; 3) FreqIN, which is a special case of Pre-ResNorm when $\lambda = 0$ in equation 3; 4) BC-ResNet-8, num SSN group=4, which is original BC-ResNet-8 in [12] with the number of subspectral normalization group of 4 [17]; 5) CP-ResNet [8], which is a residual network-based ASC model with limited receptive field size.

Table 3: **Submitted Models.** We get four variants by considering use of entire development set, device translator, model selection method, pruning ratio, bits per parameter, and number of ensemble. The symbols ‘O’ and ‘X’ imply that we utilize certain method or not, respectively. In addition, ‘Official Split’ denotes the test performance when we train each system with provided the cross-validation setup.

Model	Full Dev. data	Translator	Model Select	Prune Ratio (Conv)	Bits per Param	Ensemble	#Params (total/non-zero)	Size	Submitted		Official Split	
									Acc (%)	LogLoss	Acc (%)	LogLoss
1	teacher (O) student (X)	X	Min Var	89%	8-bit (conv) + 16-bit	2	630k/95k	121.9KB	81.06	0.56	76.99	0.72
2	teacher (w/o test set) student (w/o test set)	O	Last	89%	8-bit (conv) + 16-bit	2	630k/95k	121.9KB	86.28	0.40	75.91	0.72
3	teacher (O) student (O)	X	Last	89%	8-bit (conv) + 16-bit	2	630k/95k	121.9KB	-	-	77.46	0.72
4	teacher (O) student (X)	O	Last	84%	16-bit	X	313k/63k	122.5KB	82.17	0.51	75.10	0.77

3.2. Results and Discussion

Table 2 compares baselines with our *BC-ResNet-Mod-1* and *BC-ResNet-Mod-8* whose base number of channels c are 10 and 80 respectively in Table 1. We use fixed $\lambda = 0.1$ for Pre-ResNorm and ResNorm in experiments.

Network architecture. We compare our BC-ResNet-Mod-8 with BC-ResNet-8 [12] and CP-ResNet [8]. As shown in Table 2, BC-ResNet-Mod-8 records Top-1 test accuracy 69.5% with only one-third number of parameters compared to CP-ResNet showing 67.8% accuracy. Moreover, BC-ResNet-Mod-8 outperforms the original BC-ResNet-8 by a 1% margin with modifications.

Impact of Residual Normalization. Residual normalization plays an important role in generalization to device imbalanced dataset. First, we compare ResNorm with the baselines in BC-ResNet-Mod-1. Baselines, Global FreqNorm and FreqIN are preprocessing modules. While Global FreqNorm poorly generalizes to unseen devices S4-S6, FreqIN-based models generalize better and achieve a 5% margin compared to the Global FreqNorm. However, FreqIN shows degradation in seen devices A-S3 due to strong regularization. If we apply ResNorm as a preprocessing module (Pre-ResNorm) with a fixed $\lambda = 0.1$, the model can achieve 67.1% test accuracy. Pre-ResNorm shows higher test accuracy for seen domains A-S3 than FreqIN while maintaining good generalization for unseen devices. The results imply that ResNorm discards the unnecessary domain information by IN while keeping the useful information through the connection, $\lambda \cdot x$ in equation 3. The BC-ResNet-Mod-8 is improved by a large margin using ResNorm and achieves average test accuracy of 75.2%. Due to its regularization effect, ResNorm depresses the performance of the small model, BC-ResNet-Mod-1. In Table 2, Pre-ResNorm which uses ResNorm once at preprocessing works better than ResNorm for the small model while ResNorm works better for the large model, BC-ResNet-Mod-8.

Impact of spectrogram-to-spectrogram translation. In Fig. 1, we illustrate spectrograms translated by the real data of device a. Each spectrogram contains the characteristic of its device. We convert all training data and train our scene classification model with both original and translated datasets. As shown in Table 2, the performance variation across the devices decreases, and it indicates that training with translated data can improve the domain generalization ability of the network.

3.3. Submitted Models

To get higher accuracy, we train the network using the best method, BC-ResNet-Mod-8 with ResNorm, for a total of 300 epochs with 10 epochs of warmup. It achieves average test accuracy of 75.3%.

Also, we utilize the trained model as the teacher and train the newly initialized model by performing knowledge distillation. Then our model achieves 76.3%. We compress the model by 89% pruning, 8-bit quantization, and half-precision for convolution layers and other layers, respectively, and it achieves 75.3% as in Table 2.

Task1A allows submitting four different models. We choose various models as shown in Table 3. Our objective is to train the models performing well on the TAU Urban AcousticScenes 2021 Mobile, evaluation dataset [6] whose labels are unknown.

For models “1” and “4”, we utilize the full development set for the teacher models, but the students are trained on the original cross-validation setup. We train the teacher and student on the full development set without the cross-validation test set for model “2”. Only model “3” uses full development set for both teacher and student models. We use two methods for model selection: “Last” is the model at the end of training, and “Min Var” means the model having minimum performance differences among devices. In addition, we adopt the following variations: 1) doing 8-bit quantization and ensemble two models with different seeds or using a single half-precision model and 2) using device translator or not. *Model 1, 2, and 3* have 66.1k 8-bit nonzero for convolution layers and 29.4k 16-bit parameters for normalization, resulting in 121.9KB. *Model 4* has 48.0k nonzero parameters for convolution layers and 14.7k for others. And the model size is 122.5KB in total with 16-bit half-precision.

4. CONCLUSIONS

In this work, we design a system to achieve two goals; 1) efficient design in terms of the number of parameters and 2) adapting to device imbalanced dataset. To design an efficient acoustic scene classification model, we suggest a modified version of Broadcasting residual network [12] by limiting receptive field and using max-pool. We compress the model further by utilizing three model compression schemes, pruning, quantization, and knowledge distillation. Also, we suggest a frequency-wise normalization method, named Residual Normalization which uses instance normalization by frequency and shortcut connection to be generalized to multiple devices while not losing discriminative information. Our system achieves 76.3% test accuracy on TAU Urban Acoustic Scenes 2020 Mobile, development dataset with 315k number of parameters and the compressed version achieves 75.3% test accuracy with 89% pruning, 8-bit quantization, and knowledge distillation.

5. REFERENCES

- [1] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "DCASE 2016 acoustic scene classification using convolutional neural networks," in *Proc. Workshop Detection Classif. Acoust. Scenes Events*, 2016, pp. 95–99.
- [2] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005. IEEE, 2005, pp. 158–161.
- [3] 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.
- [4] <http://dcase.community/challenge2021/>.
- [5] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, "Low-complexity acoustic scene classification for multi-device audio: analysis of dcase 2021 challenge systems," 2021.
- [6] 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 (DCASE2020)*, 2020, submitted. [Online]. Available: <https://arxiv.org/abs/2005.14623>
- [7] S. Suh, S. Park, Y. Jeong, and T. Lee, "Designing acoustic scene classification models with CNN variants," DCASE2020 Challenge, Tech. Rep., June 2020.
- [8] K. Koutini, H. Eghbal-zadeh, M. Dorfer, and G. Widmer, "The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification," in *EUSIPCO*. IEEE, 2019, pp. 1–5.
- [9] 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," DCASE2020 Challenge, Tech. Rep., June 2020.
- [10] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, "Integrating the data augmentation scheme with various classifiers for acoustic scene modeling," DCASE2019 Challenge, Tech. Rep., June 2019.
- [11] S. S. R. Phaye, E. Benetos, and Y. Wang, "Subspectralnet—using sub-spectrogram based convolutional neural networks for acoustic scene classification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 825–829.
- [12] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," *CoRR*, vol. abs/2106.04140, 2021.
- [13] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016.
- [14] H. Nam and H. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," in *NeurIPS*, 2018, pp. 2563–2572.
- [15] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*. IEEE Computer Society, 2017, pp. 1510–1519.
- [16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [17] S. Chang, H. Park, J. Cho, H. Park, S. Yun, and K. Hwang, "Subspectral normalization for neural audio data processing," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 850–854.
- [18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [19] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.
- [20] J. Kim, K. Yoo, and N. Kwak, "Position-based scaled gradient for model quantization and pruning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 20415–20426.
- [21] J. Kim, M. Hyun, I. Chung, and N. Kwak, "Feature fusion for online mutual knowledge distillation," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 4619–4625.
- [22] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR (Poster)*. OpenReview.net, 2018.
- [23] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *INTER-SPEECH*. ISCA, 2019, pp. 2613–2617.
- [24] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.
- [25] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *ICLR (Poster)*. OpenReview.net, 2017.