# DCASE2023 TASK1 SUBMISSION: DEVICE SIMULATION AND TIME-FREQUENCY SEPARABLE CONVOLUTION FOR ACOUSTIC SCENE CLASSIFICATION

Technical Report

*Yiqiang Cai*[1], *Minyu Lin*[1], *Chenyang Zhu*[2], *Shengchen Li*[1], *Xi Shao*[2]

[1] Xi'an Jiaotong-Liverpool University, School of Advanced Technology, Suzhou, China,
{yiqiang.cai21, minyu.lin20}@student.xjtlu.edu.cn, shengchen.li@xjtlu.edu.cn
[2] Nanjing University of Posts and Telecommunications,
College of Telecommunications and Information Engineering, Nanjing, China,
shaoxi@njupt.edu.cn, chenyangzhu2018@163.com

## ABSTRACT

The task 1 of DCASE 2023 Challenge incorporates a weighted average ranking of accuracy and complexity, which encourages participants to build efficient systems for acoustic scene classification (ASC). In this report, we propose TF-SepNet, a low-complexity ASC model based on Time-Frequency Separable Convolution. Our network architecture consists of a series of separable convolutional layers that exploit time and frequency domains. We also improve the performance of ResNorm by adding a few learnable parameters. Furthermore, knowledge distillation is employed to transfer knowledge from large model to smaller model. Additionally, device simulation is introduced for data augmentation in the device domain. Overall, we evaluate the performance of our model on the DCASE 2023 Task 1 development dataset following the official cross-validation setup and achieve a classification accuracy of $53.9\%$ with $6.83K$ parameters and $1.65M$ MACs.

***Index Terms—*** Acoustic scene classification, efficient neural network, device simulation, knowledge distillation

## 1. INTRODUCTION

Acoustic scene classification (ASC) [1] has gained significant attention in recent years due to its wide range of applications, such as surveillance, smart homes, and environmental monitoring. Acoustic scenes are commonly diffused with a large amount of mixed information like the sounds of people talking, car driving, noise etc. The Task 1 of DCASE (Detection and Classification of Acoustic Scenes and Events) Challenge [2] is a well-known benchmark for evaluating ASC methods, focusing on scene classification such as underground stations, street traffic or public squares.

The dataset used for this task consists of recordings of 10 distinct acoustic scenes that were captured across 12 cities using various devices. Additionally, partially synthesized data was created from the original recordings. To facilitate the challenge, each segment from the dataset was reduced in duration from 10 seconds to 1 second. This makes the task more challenging since the classifier must now predict the acoustic scene based on much less information, reducing the amount of informative features available for analysis. Furthermore, the 2023 DCASE Challenge Task 1 introduces a new evaluation metric that considers both accuracy and complexity, motivating participants to develop a comprehensive approach to using low-computational resources rather than solely focusing on meeting memory and MAC limits. This challenge presents an exciting opportunity to explore innovative approaches for developing high-performing ASC models with low computational requirements.

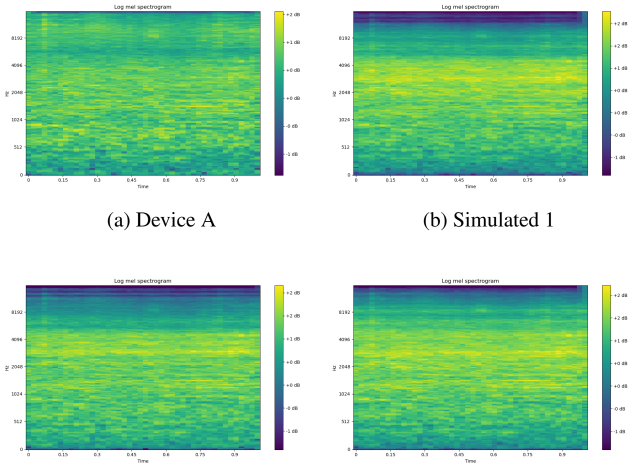
In this report, we present our approach to solving the Task 1 of DCASE 2023 Challenge. Firstly, our proposed solution, TF-SepNet, is a low-complexity neural network architecture based on Time-Frequency Separable Convolution that leverages the property of convolution operation in time and frequency domains. Moreover, a few learnable parameters are added to the Residual Normalization [3] layer to further enhance the performance. Secondly, the device-domain generalization ability of our model is improved by introducing device simulation for data augmentation, which employs impulse responses from the MicIRP dataset [4] to simulate the impact of diverse recording devices on recorded sounds. Thirdly, different strategies of data augmentation are applied to address the overfitting problem, i.e. Mixup [5] and Freq-MixStyle [6]. Finally, knowledge distillation and quantization are the strategies for model compression.

Our work contributes to the ASC literature by demonstrating the effectiveness of low-complexity neural network architectures, knowledge distillation, and innovative data augmentation techniques for achieving high accuracy on the DCASE 2023 Task 1 dataset. Through our comprehensive analysis and experimental evaluations, we provide insights into the impact of different components of our model, which can guide future research in the area of ASC.

## 2. DATA PREPROCESSING AND AUGMENTATION

### 2.1. Dataset

The TAU Urban Acoustic Scene 2022 Mobile development dataset [7] is a subset of the larger TAU Urban Acoustic Scenes 2022 dataset, consisting of recordings captured using mobile devices in urban environments. The dataset includes 230,350 audio clips, each with a duration of 1 seconds and a hard label of an acoustic scene. There are totally 10 different acoustic scene categories including airport, bus, metro, metro station, park, public square, street pedestrian, street traffic, tram, and urban park. The recordings were captured across several cities around the world and using a wide range of mobile devices. While the dataset has a balanced distribution of samples across each of the acoustic scene categories, it is worth



(a) Device A

(b) Simulated 1

(c) Simulated 2

(d) Simulated 2

Figure 1: **Examples of device simulation.** (a) is extrated from a recording of real device A. (b), (c) and (d) are simulated by convolving recording A with 3 different impulse responses.

noting that there is an imbalanced number of samples recorded by diverse devices. The main recording device A was a Zoom F8 recorder with binaural microphones, which contributes to 73% of the data. The simulated devices were synthesized by processing data from device A. In addition, some devices of evaluation dataset are unseen in development dataset.

### 2.2. Feature Extraction

Following the setup of [6], all audio segments are resampled with 32kHz sample rate. The features are extracted by using Short Time Fourier Transform with window size of 2048 and hop size of 744. A Mel-scaled filter bank is then applied with 128 frequency bins. As a result, the shape of an input feature is $1 \times 128 \times 44$.

### 2.3. Device Simulation

Device simulation is an audio-based augmentation method, which aims to simulate recordings from one device to other devices. Following the official setup, a number of random recordings from device A is convolved with selected impulse response from the MicIRP dataset [4] to simulate a new device. A few examples are visualized as shown in Fig. 1. The operation can be denoted by Eq. (1),

$$\mathcal{F}(a, v) = (a * v)_n = \sum_{m=-\infty}^{\infty} a_m \cdot v_{n-m} \tag{1}$$

where $a$ is the original recording of device A and $v$ is the selected impulse response, $m$ and $n$ represent the indices or time samples of the signals being convolved.

In the experiments, 32 impulse response files are used for device simulation. For each impulse response, 7500 samples of device A is randomly selected to create a new device. Therefore, the total number of augmented data is 240,000.

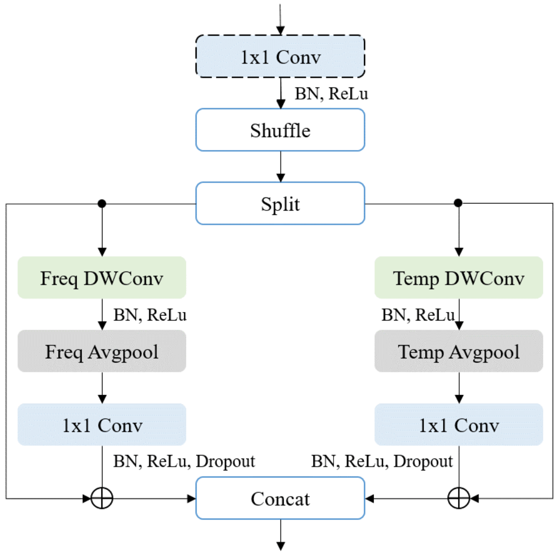


Figure 2: **Time-Frequency Separable Convolution.**

### 2.4. Mixup and Freq-MixStyle

Mixup [5] and MixStyle [8] are two popular feature-based augmentation techniques for creating synthetic samples. Both techniques have been shown to be effective in improving the robustness and accuracy of deep neural networks. Mixup generates a new training sample by linearly interpolating two random examples and their corresponding labels, while MixStyle adapts the style of one example to another using a learnable style transfer module. Freq-MixStyle is introduced by [6] for the ASC task, which normalizes the frequency bands instead of channels. The combination of Mixup and Freq-MixStyle is applied to the input in our experiments.

The mixing coefficient of Mixup and Freq-MixStyle $\alpha$ is both set to 0.3, while the probability of applying Freq-MixStyle $p$ is experimented with different values.

## 3. ASC MODEL

### 3.1. Network Architecture

The proposed CNN architecture, called TF-SepNet, is designed to be a low-complexity model that can efficiently extract discriminative features from audio recordings. TF-SepNet is based on the Time-Frequency Separable Convolution, as illustrated in Fig. 2, which separates the temporal and spectral information in the input signals and processes them independently using two sets of convolutional filters. The first $1 \times 1$ convolutional layer is used when there is a need to expand or shrink the number of channels. Subsequently, a shuffle layer [9] is added to establish connections between the channels, and the shuffle group is set to a quarter of the number of input channels. Following this, the channels are evenly divided into two halves and separately processed in the frequency and time domains using separable operations, then residual identities are respectively added to the outputs. Finally, the resulting feature maps are concatenated together.

Specifically, as shown in Tab. 1, the architecture of TF-SepNet is adopted from BC-ResNet [10] but the BC-ResBlocks are replaced

| BLOCK | N | SHAPE |
|---|---|---|
| Input feature | - | $1 \times F \times T$ |
| Conv 5×5 | - | $2C \times F/2 \times T/2$ |
| TF-SepConv | 2 | $C \times F/2 \times T/2$ |
| MaxPool 2×2 | - | $C \times F/4 \times T/4$ |
| TF-SepConv | 2 | $1.5C \times F/4 \times T/4$ |
| MaxPool 2×2 | - | $1.5C \times F/8 \times T/8$ |
| TF-SepConv | 2 | $2C \times F/8 \times T/8$ |
| TF-SepConv | 3 | $2.5C \times F/8 \times T/8$ |
| Conv 1×1 | - | $10 \times F/8 \times T/8$ |
| AvgPool | - | $10 \times 1 \times 1$ |

Table 1: **Architecture of TF-SepNet**. N denotes the number of blocks. $C$, $F$, and $T$ respectively represent the number of channel, frequency bins, and time clips.

by TF-SepConv blocks. The model starts with a 5x5 convolution at the beginning that downsamples using a 2x2 stride. Afterward, it has a total of 9 TF-SepConv blocks and two 2x2 maxpool layers with 2x2 stride. Lastly, a 1x1 convolution is performed prior to global average pooling which allows the model to classify the outputs into 10 classes. The number of output channels $C$ is leveraged as a hyper-parameter to adjust the complexity of model.

### 3.2. Adaptive Residual Normalization

As illustrated in Eq. (2), frequency-instance normalization (FreqIN) [3] generalizes the features on the device domain by applying instance normalization (IN) in the frequency dimension. Moreover, residual normalization (ResNorm) [3] adds an identity path to FreqIN with a hyper-parameter $\lambda$ for compensating the information loss, as shown in Eq. (3).

$$FreqIN(x) = \frac{x - \mu_{nf}}{\sqrt{\sigma_{nf}^2 + \epsilon}} \tag{2}$$

$$ResNorm(x) = \lambda \cdot x + FreqIN(x) \tag{3}$$

Here the input feature $x \in \mathbb{R}^{n \times c \times f \times t}$ is a 4-dimensional vector with batch size, channels, frequency bins and temporal clips. $\mu_{nf}$ and $\sigma_{nf}$ separately indicate the mean and standard deviation of input on $n$ and $f$ dimensions. $\epsilon$ is an extremely small constant for numerical stability.

Inspired by [11], we introduces adaptive residual normalization, as shown in Eq. (4), by adding trainable parameters to control the trade-off between identity and FreqIN. By doing so, the normalization behavior can be adaptively adjust based on the characteristics of the data and the requirements of the task.

$$AdaResNorm(x) = (\rho \cdot x + (1 - \rho) \cdot FreqIN(x)) \cdot \gamma + \beta \tag{4}$$

Here $\rho$, $\gamma$ and $\beta$ are trainable parameters for balancing, scaling and shifting. Adaptive residual normalization is inserted after the first convolution layer and each stage of TF-SepConv blocks.

### 3.3. Knowledge Distillation

In this work, knowledge distillation is implemented as a means of transferring knowledge from a larger model (referred to as the ”teacher” model) to a smaller model (referred to as the ”student” model). Knowledge distillation has proven to be an effective approach for model compression and improving the generalization capabilities of smaller models in ASC tasks [6] [12].

During the training process, the student model not only learns from the ground truth labels but also takes advantage of the soft targets generated by the teacher model. As shown in Eq. (5), the loss function consists of label loss and distillation loss. The label loss is the cross entropy between student output and ground truth labels. The distillation loss refers to the KL divergence between student output and teacher soft targets. These soft targets represent the teacher model’s output probabilities or logits, which provide more nuanced information than simple one-hot labels. By considering the soft targets, the student model can learn from the teacher model’s knowledge about the underlying relationships and uncertainties in the data.

$$L = L_{label} + \lambda L_{dist} \tag{5}$$

In this work, $\lambda$ is set to 20 and the tempereture is set to 5. TF-SepNet with $C = 160$ is chosen as the teacher model while only TF-SepNet with $C = 12$ is used as the student model in this work.

## 4. TRAINING SETUP

We train the models for 200 epoch using Adam optimizer with default settings and batch size to 32. The learning rate is scheduled to linearly increase from 0 to 0.01 in ten epochs as a warmup [13], then decay to 0 with cosine annealing [14] for the rest of epochs. $\alpha$, $p$ of Freq-MixStyle and dropout rate $d$ are adjusted to improve the overfitting problem. After training, Post-Training Static Quantization in Pytorch [15] is implemented to quantize the parameters of model to INT8 data type. The combination of Convolution, Batch Norm and ReLu layers are fused to improve accuracy.

## 5. RESULTS AND SUBMISSIONS

The results are shown in Tab. 3. Our TF-SepNet ($C = 40$) outperforms BC-ResNet ($C = 40$) with fewer parameters. The Adaptive ResNorm improves $0.3\%$ of accuracy by introducing $2\%$ parameters. Device simulation greatly increase the generalization ability of models in the device domain, which makes the accuracy of the model in the unseen domains reach a level comparable to that in the seen domains. With $C$ being set to 12, 20, 40, 60 and 160, we get models with different complexities. Knowledge distillation leverages the expertise and generalization capabilities of TF-SepNet ($C = 160$), resulting in improved performance of TF-SepNet ($C = 12$). Submissions are shown in Tab. 2, and all submission models have been trained on the whole development dataset.

| ID | Model | KD | $p$ |
|---|---|---|---|
| 1 | TF-SepNet, $C$=12 | √ | 0.5 |
| 2 | TF-SepNet, $C$=12 | × | 0.5 |
| 3 | TF-SepNet, $C$=20 | × | 0.6 |
| 4 | TF-SepNet, $C$=40 | × | 0.7 |

Table 2: Submissions

| Model | Seen | | | | | | Unseen | | | Performance | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | S1 | S2 | S3 | S4 | S5 | S6 | Acc/% | MACs/M | Param/K |
| BC-ResNet, $C$=40 + ResNorm | 66.0 | 52.4 | 56.7 | 54.2 | 55.1 | 59.0 | 50.2 | 50.4 | 43.3 | 54.2 | 10.23 | 87.04 |
| TF-SepNet, $C$=40 + ResNorm | 67.9 | 57.8 | 63.5 | 56.4 | 52.4 | 57.5 | 53.4 | 55.5 | 50.6 | 57.2 | 10.22 | 53.19 |
| + AdaResNorm (1) | 67.3 | 58.4 | 64.1 | 57.0 | 52.4 | 56.5 | 54.8 | 55.0 | 51.8 | 57.5 | 10.22 | 54.27 |
| + Device Simulation (2) | 76.5 | 60.3 | 68.4 | 63.2 | 61.6 | 67.1 | 61.9 | 61.4 | 58.6 | 64.3 | 10.22 | 54.27 |
| TF-SepNet, $C$=12 + (1)(2) | 62.1 | 49.8 | 56.4 | 47.7 | 50.4 | 53.8 | 52.5 | 49.9 | 44.8 | 51.9 | 1.65 | 6.83 |
| + Knowledge Distillation | 64.4 | 52.9 | 58.8 | 52.0 | 48.8 | 55.7 | 53.5 | 53.1 | 46.1 | 53.9 | 1.65 | 6.83 |
| TF-SepNet, $C$=20 + (1)(2) | 69.6 | 56.0 | 61.0 | 54.3 | 54.5 | 59.5 | 56.5 | 56.1 | 50.3 | 57.5 | 3.42 | 15.89 |
| TF-SepNet, $C$=60 + (1)(2) | 79.2 | 62.1 | 69.2 | 64.6 | 61.1 | 68.8 | 64.6 | 66.2 | 59.5 | 66.2 | 20.39 | 115.15 |
| TF-SepNet, $C$=160 + (1)(2) | 89.4 | 69.8 | 78.0 | 77.0 | 74.0 | 79.3 | 76.4 | 75.2 | 69.2 | 76.5 | 242.18 | 757.05 |

Table 3: **Results.** These results are obtained by training 100 epochs for time saving while submissions by training 200 epochs.

## 6. CONCLUSION

In this report, we have presented our approach for solving Task 1 of the 2023 DCASE Challenge, which focuses on the efficiency of ASC system. We propose TF-SepNet, a low-complexity network based on Time-Frequency Separable Convolution, and outperform the state-of-the-art ASC systems. In addition, we introduced device simulation to augment data in the device domain. Moreover, we employed knowledge distillation to transfer knowledge from a larger teacher model to a smaller student model.

## 7. ACKNOWLEDGEMENT

This project is supported partly by the National Natural Science Foundation of China (No: 62001038) and Gusu Innovation and Entrepreneurship Leading Talents Programme (No: ZXL2022472).

## 8. REFERENCES

[1] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[2] 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," 2022.

[3] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design," DCASE2021 Challenge, Tech. Rep., June 2021.

[4] "MicIRP: IR data for vintage microphones," Personal Blog, accessed on 2023.05. [Online]. Available: http://micirp.blogspot.com/?m=1

[5] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[6] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to dcase22: Distilling knowledge for low-complexity convolutional neural networks from a patchout audio transformer," DCASE2022 Challenge, Tech. Rep., June 2022.

[7] 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, pp. 56–60.

[8] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," *arXiv preprint arXiv:2104.02008*, 2021.

[9] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[10] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," *arXiv preprint arXiv:2106.04140*, 2021.

[11] H. Nam and H.-E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[12] T. Morocutti and D. Shalaby, "Receptive field regularized CNNs with traditional audio augmentations," DCASE2022 Challenge, Tech. Rep., June 2022.

[13] 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.

[14] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.