2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)

# Enhancing Spectrogram for Audio Classification Using Time-Frequency Enhancer

Haoran Xing* Shiqi Zhang* Daiki Takeuchi† Daisuke Niizumi† Noboru Harada† Shoji Makino*
* Waseda University, Japan
E-mail: haoranxing@suou.waseda.jp, tiosisai@gmail.com, s.makino@waseda.jp
† NTT Corporation, Japan
E-mail: noboru.harada.pv@hco.ntt.co.jp, daisuke.niizumi@ntt.com, daiki.takeuchi.ux@hco.ntt.co.jp

***Abstract*—It is challenging to deploy Transformer-based audio classification models on common terminal devices in real situations due to their high computational costs, increasing the importance of transferring knowledge from the larger Transformer-based model to the smaller convolutional neural networks (CNN)-based model via knowledge distillation (KD). Since an audio spectrogram can be regarded as an image, image-based models with CNN-based structures are used as the aforementioned smaller model for KD in several studies. However, the physical meanings of spectrograms differ from that of images in general. This fact possibly leads to the issue that the image-based model may not effectively extract features from a pure spectrogram. Thus, improving the spectrogram can help these models perform better on audio classification tasks. To implement our hypothesis, we propose a new Time-Frequency Enhancer (TFE), which is designed to learn how to enhance input spectrograms to make them effective for audio classification. In addition, we also propose TFE-ENV2, which extends EfficientNetV2 (ENV2), an image-based backbone model. To verify the effectiveness of the proposed method, we compare the performance between the original ENV2 and the proposed TFE-ENV2. In our experiments, the proposed TFE-ENV2 outperformed the original ENV2 on the ESC-50 and Speech Commands V2 datasets, demonstrating that the proposed TFE enhances spectrograms to assist image-based models in audio classification.**

## I. Introduction

Audio classification task is aimed at classifying various audios and understanding the acoustic environment [1]. It has been used to assist speech enhancement [2], recognizing speech commands [3], and automatic speech recognition (ASR) [4], etc.

Convolutional Neural Network (CNN)-based models [5] have built-in inductive biases, which enable them to learn from limited audio data with small sizes [6, 7]. Recently, researchers have integrated image-based models with CNN architectures into audio classification tasks, leveraging their superior performance in the image domain to classify the spectrogram, which could be considered as the generalized image. At the same time, although some Transformer-based models [8–10] have shown better performance than CNN recently by incorporating techniques such as attention mechanisms [11], they typically have larger sizes and require higher computational costs, making them impractical for low-resource devices. For more usability on terminal devices in practical circumstances, previous studies have done research to transfer knowledge from larger Transformer-based models to smaller CNN-based models using knowledge distillation (KD) [12].

To enable both parameter/computationally efficiency and a high-performance model for audio classification tasks, this study focuses on training image-based models with CNN architectures via transferring knowledge from a large, high-performance Transformer-based model using KD. Both the Transformer-based model and the image-based model take the spectrogram as input.

Instead of feeding the original spectrogram to image-based models, we think that improving the spectrograms can help these models perform better on audio classification tasks since spectrograms and common images have different physical meanings. Values in spectrograms describe the energy distribution of the audio signal in the time-frequency domain, while images do not have this physical meaning.

To address the issue, we propose a new Time-Frequency Enhancer (TFE) to enhance the features of spectrograms in the time-frequency domain. The TFE consists of several learnable Enhancer Blocks, which are responsible for extracting features and generating a mask used for enhancing the spectrogram. The mask assigns importance weights to the features in the spectrogram, resulting in enhanced spectrograms. The TFE module is trained jointly with the image-based model using KD. While the TFE acts as a front end to provide enhanced spectrograms, the image-based model serves as the backbone. This joint optimization enables them to work together and achieve improved performance in audio classification. The TFE learns to enhance the spectrogram, thereby enhancing the audio classification performance.

Cross-Model Knowledge Distillation (CMKD) method [13] has been proven to be effective for transferring knowledge on audio classification. Besides, the Audio Spectrogram Transformer (AST) model has outstanding performance on audio classification [8] and EfficientNet (EN) models [14, 15] have been proven to perform well on audio classification in previous studies [9, 13]. Among all EN models, EfficientNetV2 (ENV2) [16] has been proven to outperform previous work on the ImageNet [17] dataset. The employment of the Fused-Mobile Convolution (Fused-MBConv) blocks [18] in the early stages of ENV2 can reduce training time and provide better efficiency for image classification tasks. Thus, to further improve the performance of audio classification, we transfer knowledge

979-8-3503-0067-3/23/$31.00 ©2023 IEEE

from an AST model with good performance to the audio classification model with the backbone of ENV2 through CMKD.

The remainder of the paper is organized as follows: Section II provides an overview of the related work in this paper. In Section III, we introduce the proposed method and discuss its architecture. Experimental settings and results are presented in Section IV. Finally, in Section V, we conclude the paper.

## II. Related Work

### A. Audio Spectrogram Transformer

Audio Spectrogram Transformer (AST) [8] is a Transformer-based model for audio classification tasks that outperformed other CNN-based models at the time. Unlike CNN-based models for audio classification, AST works purely based on the Self-Attention mechanism [19] to capture long-range global information from spectrograms. However, AST is large and needs large-scale datasets for training, which leads to high training costs and high computational complexity. Thus, it is difficult to deploy AST on typical terminal devices in practical settings.

### B. EfficientNet Model for Audio Classification

EfficientNet (EN) is a CNN-based model originally proposed for image classification tasks. EN works based on the structure of multiple Mobile Convolution (MBConv) blocks [20]. MBConv is a residual block with Squeeze-and-Excitation (SE) module [21], which is designed to capture channel-wise dependencies and generate importance weights for different channels. Several studies have used AudioSet [22] to pretrain EN to further improve audio classification performance. However, direct pretraining EN on AudioSet is time-consuming and leads to high training costs. Transferring knowledge from pretrained models to EN is important for this reason.

### C. Cross-Model Knowledge Distillation

Cross-Model Knowledge Distillation (CMKD) [13] method is aimed at transferring knowledge between audio classification models with different architectures using KD, which is a common method allowing small student models to learn from large teacher models [23]. In the original paper of CMKD, KD is employed to transfer knowledge from a pretrained AST to an EN. It has been demonstrated that EN performs better after learning from pretrained AST. Therefore, CMKD is a practical method to obtain an audio classification model with good performance and a smaller size.

### D. Learnable Frontend for Audio Classification

The related work Learnable Frontend (LEAF) [24] for audio classification is designed as a replacement for mel-filterbank. It learns audio feature extraction and converts the audio signal from the time domain into the time-frequency domain. However, due to KD being used for training, the input for the image-based CNN model is expected to be the same as that for the AST model, which is the Log Mel Spectrogram, a typical representation of audio signals in the time-frequency domain. Thus, enhancing the spectrogram in the time-frequency domain is required for image-based models rather than feature extraction from the time domain to the time-frequency domain.

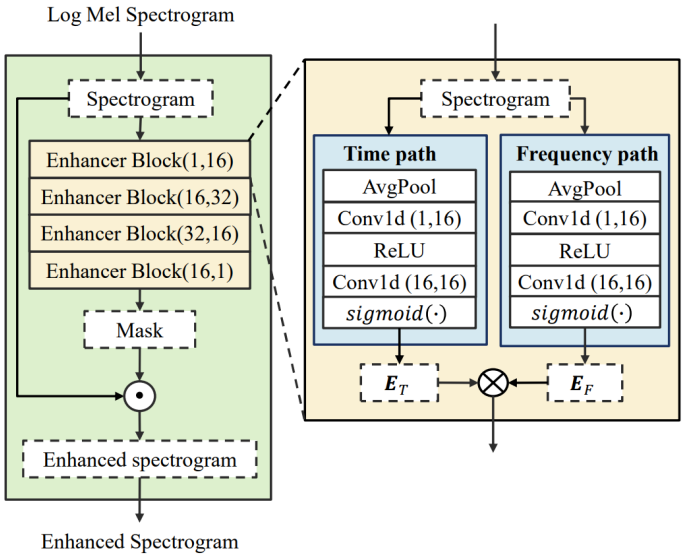


Fig. 1. Illustration for the architecture of the proposed TFE module

## III. Proposed Method

We propose a new Time-Frequency Enhancer (TFE), which is intended to enhance the original spectrogram and provide the enhanced spectrogram as input for image-based models. In addition, we propose TFE-ENV2, in which we apply TFE as a front end to ENV2 for audio classification.

### A. Time-Frequency Enhancer (TFE)

We propose the TFE module, inspired by Time-Frequency Attention (TFA) [25], which was designed for the speech enhancement task. TFA is a lightweight attention mechanism that provides the mask for the spectrogram in the time-frequency domain. It can be regarded as an Enhancer Block for the spectrogram, which employs two independent feature extractors with the same structures for the time frame path and the frequency bin path in the spectrogram, respectively. Both extractors consist of the average pooling layer (AvgPool), 1D convolution layer (Conv1d), ReLU activation function (ReLU), and Sigmoid activation function ($sigmoid(\cdot)$). In each path, at first, the 2D spectrogram is compressed into a 1D feature map through the average pooling operation. Then the 1D convolution layer is employed to extract features from the 1D feature map. Each extractor outputs an enhanced 1D feature vector. Two 1D vectors from the time path and frequency path would be used to generate a 2D mask for the input spectrogram by matrix multiplication, which can be written as

$$\mathbf{M} = \mathbf{E_T} \otimes \mathbf{E_F}, \tag{1}$$

where $\mathbf{M}$ represents the 2D mask, $\mathbf{E_T}$ represents the enhanced feature vector from the time path, $\mathbf{E_F}$ represents the enhanced

feature vector from the frequency path, and $\otimes$ represents matrix multiplication. The architecture of the Enhancer Block is shown in the yellow block (right) of Fig. 1.

The structure of the proposed TFE module, made up of multiple Enhancer Blocks, is shown in the green block (left) of Fig. 1. In the remaining paper, we use TFE(**n**) to represent the TFE module, which consists of **n** Enhancer Blocks. The TFE module in Fig. 1 includes 4 Enhancer Blocks, denoted as TFE(4). Compared with TFA, which uses only one Enhancer Block for spectrogram enhancement, the proposed TFE(**n**) module works based on **n** Enhancer Blocks to enhance the spectrogram in multiple channels. By deepening the network structure, the proposed TFE(**n**) should learn rich features that enable the model to capture more helpful information from the spectrogram. The TFE(**n**) module takes the original Log Mel Spectrogram as the input and generates the mask through **n** Enhancer Blocks. After that, the mask and input spectrogram are combined into the enhanced spectrogram using the element-wise product, which is expressed as

$$\widetilde{\mathbf{S}} = \mathbf{S} \odot \mathbf{M}, \tag{2}$$

where $\mathbf{S}$ represents the original spectrogram, $\mathbf{M}$ represents obtained mask, $\widetilde{\mathbf{S}}$ represents the enhanced spectrogram, and $\odot$ represents the element-wise product.

The proposed TFE module produces an attention mask to enhance the original spectrogram by multiplying two feature vectors, which are output from the time path and frequency path in the spectrogram, respectively. These two vectors emphasize the important points in these two paths based on the energy distribution of the spectrogram, respectively. As a result, the attention mask is intended to emphasize the important points in the whole spectrogram and provide significance weights for the spectrogram. Thus, as the combination of the attention mask and the original spectrogram, the enhanced spectrogram should provide more helpful features for audio classification.

### *B. EfficientNetV2 with TFE module (TFE-ENV2)*

In this work, we propose a combined structure for audio classification that integrates EfficientNetV2 with the TFE module. This structure, named TFE-ENV2, incorporates the TFE module for spectrogram enhancement and the ENV2 module for classification.

Fig. 2(a) illustrates the architecture of the original ENV2 while Fig. 2(b) illustrates the architecture of the proposed TFE-ENV2. The original ENV2 consists of a 2D convolutional layer (with a kernel size of 3×3), Fused-MBConv blocks, MBConv blocks, and a classifier, which is the same as that employed for image domain [16].

The proposed TFE-ENV2 structure utilizes a Log Mel Spectrogram as input. The TFE module serves as the front end, while the ENV2 module functions as the backbone for audio classification. This combination is chosen based on specific advantages and expected improvements. By incorporating the TFE module, we aim to enhance the feature extraction process, leading to improved classification performance compared to the original ENV2.

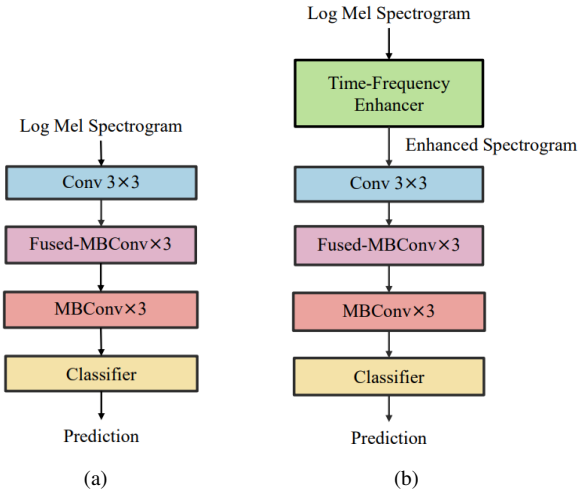


Fig. 2. Illustration for architectures of (a) the original ENV2 and (b) the proposed TFE-ENV2.

## IV. Experiment

We validated the proposed method on ESC-50 [26] dataset and Speech Commands V2 [27] dataset.

To enhance audio classification performance, we employed the CMKD method to transfer knowledge from the AST model to the TFE-ENV2. And to ensure consistency in our approach, we reused the experimental settings of CMKD as described in the previous work [13] as shown in Table I.

### *A. Datasets*

The experiments were conducted on two labeled datasets: ESC-50 and Speech Commands V2. The ESC-50 dataset is made up of 2,000 5-second environmental sounds organized into 50 classes. We followed the 5-fold cross-validation and recorded the mean accuracy. The speech Commands V2 dataset contains 105,829 different 1-second recordings divided into 35 classes. It is divided into a training set, a validation set, and an evaluation set, made up of 84,843, 9,981, and 11,005 samples, respectively. We followed the 35-class classification task and recorded the accuracy of the evaluation set.

### *B. Experimental Setup*

To confirm the effectiveness of the proposed TFE, we conducted ablation experiments to compare the original ENV2 with the proposed TFE(**n**)-ENV2. We assessed three TFE models with varying numbers of Enhancer Blocks. **n** was adjusted to 1, 2, and 4 in our experiments. The objective was to evaluate the impact of different TFE configurations. TFE(1)-ENV2 is made up of Enhancer Block(1,1), TFE(2)-ENV2 is made up of Enhancer Block(1,32), and Enhancer Block(32,1). For a fair comparison, all student models were trained with the same training settings and learned from the same AST model. All experiments were repeated 5 times and the mean accuracy was reported.

**Model settings:** On the ESC-50 dataset, AST, the teacher model, was initialized by the weights pretrained on two

datasets (ImageNet and AudioSet) and then fine-tuned for 5 epochs. While on the Speech Commands V2 dataset, it was only pretrained on ImageNet and also fine-tuned for 5 epochs. Then, the fine-tuned models were used for knowledge distillation. The ENV2 was initialized with weights pretrained on ImageNet, serving as the backbone for audio classification. In contrast, the proposed TFE(**n**) module did not undergo any pretraining.

**Data processing and data augmentation:** For the ESC-50 dataset, to keep consistency, we followed the same data processing and data augmentation recipes as described in the CMKD method in previous work [13]. The teacher model was fine-tuned with SpecAugment [28] (frequency and time masking). After that, student models were trained using SpecAugment, random time-shift (rolling spectrograms in the time axis), random noise (generating distribution to spectrograms), and label smoothing (smoothing one hot label) [29]. For the Speech Commands V2 dataset, we followed the same data augmentation recipes as those used for the original AST in previous work [8]. The teacher model was trained with SpecAugment and mix-up (randomly mixing up two spectrograms and labels) [30]. After that, student models were trained using the same method. Input for the teacher model and student models were spectrograms with the same data augmentations during training, while there was no data augmentation during testing.

**Training settings:** All the details for implementation in experiments are shown in Table I. For ESC-50, to keep consistency, student models were trained with the same training settings used for the AST model in previous work [8], except for the initial learning rate, learning scheduler, and epochs. For ESC-50, the learning rate decreased by a factor of 0.85 every 5 epochs. For Speech Commands V2, the learning rate decreased by a factor of 0.85 for every epoch after the first 5 epochs. The Adam optimizer was used for both datasets.

TABLE I
TRAINING SETTINGS ON ESC-50 AND SPEECH COMMANDS V2

| Settings | ESC-50 | Speech Commands V2 |
|---|---|---|
| Spectrogram size (Time frames/Frequency bins) | 512/128 | 128/128 |
| Initial learning rate | 1e-4 | 5e-4 |
| Batch size | 48 | 128 |
| Epochs | 100 | 30 |
| Loss for ground truth | Cross Entropy | Brainy Cross Entropy |
| SpecAugment (Time/Frequency) | (96/24) | (48/48) |
| Random noise on spectrogram | U(0,0.05) | U(0,0.1) |
| Random time-shift | ±10 time frames | ±10 time frames |
| Mix-up ratio | 0 | 0.6 |
| Label smoothing factor | 0.1 | 0 |

**Knowledge distillation:** Distillation loss used in training can be written as

$$\mathcal{L} = \lambda \mathcal{L}_g(\delta(Z_S), y) + (1-\lambda)\mathcal{L}_d(\delta(Z_S), \delta(Z_T/\tau)), \quad (3)$$

where $Z_S$ and $Z_T$ present the logits from the student model and teacher model, respectively. $\mathcal{L}_g$ and $\mathcal{L}_d$ are losses for ground truth and distillation, respectively. $y$ represents ground-truth labels. $\lambda$ represents the balancing coefficient. $\delta$ is the activation function. $\tau$ is KD temperature. Cross-entropy (CE) loss and Kullback-Leibler divergence loss were used for $\mathcal{L}_g$ and $\mathcal{L}_d$, respectively. We fixed balancing coefficient $\lambda = 0.5$ and KD temperature $\tau = 5$. Soft-max was used as the activation function $\delta$.

### C. Results

TABLE II
RESULTS FOR ABLATION EXPERIMENTS ON THREE TFE-ENV2 MODELS

| Model Name | Parameters | Accuracy | |
|---|---|---|---|
| | | ESC-50 | Speech Commands V2 |
| TFE(1)-ENV2 | 8.75M | 89.42% (±0.43) | 97.78% (±0.04) |
| TFE(2)-ENV2 | 8.77M | 89.23% (±0.27) | 97.79% (±0.09) |
| TFE(4)-ENV2 | 8.82M | **89.67%** (±0.46) | **97.85%** (±0.04) |

Table II shows the results of ablation experiments for TFE(**n**)-ENV2 models. According to results in Table II, for the Speech Commands V2 dataset, the accuracy increased as the number of Enhancer Blocks in TFE(**n**)-ENV2 increased. Compared with TFE(1)-ENV2, the accuracy of TFE(**n**)-ENV2 increased by 0.01% and 0.07% when **n** is set to 2 and 4, respectively. While for ESC-50, TFE(1)-ENV2 performed better than TFE(2)-ENV2. One possible reason might be that there was no pretraining for the TFE(**n**) module, so the increasing number of Enhancer Blocks caused the increase in unpretrained parameters, which need more data for training. As a result, TFE(**n**)-ENV2 models might overfit on the ESC-50 dataset, which has a smaller data size than Speech Commands V2. Thus, we believe that the trend of the accuracy of TFE(**n**)-ENV2 on the Speech Commands V2 is more deserving of reference.

TABLE III
COMPARISONS FOR THE PERFORMANCE BETWEEN AST, ENV2, AND THE BEST TFE-ENV2

| Model Name | Parameters | Accuracy | |
|---|---|---|---|
| | | ESC-50 | Speech Commands V2 |
| AST | 85.70M | 95.85% | 97.46% |
| ENV2 | 8.74M | 88.63% (±0.34) | 97.74% (±0.07) |
| TFE(4)-ENV2 | 8.82M | **89.67%** (±0.46) | **97.85%** (±0.04) |

Table III shows the comparison of the performance of AST, ENV2, and the best TFE-ENV2. According to the results, the accuracy of TFE(4)-ENV2, which is the best of all, increased by 1.04% and 0.11% compared with the original ENV2 on ESC-50 dataset and Speech Commands V2 dataset, respectively. Besides, all the proposed TFE(**n**)-ENV2 models outperformed the original ENV2 model. What's more, on Speech Commands V2, the accuracy of TFE(4)-ENV2 got almost 0.4% higher than AST with only 10.29% of its parameters. All these results demonstrated the effectiveness of the proposed TFE.

TABLE IV
PERFORMANCE OF ENV2 AND TFE(4)-ENV2 WHEN CMKD WAS SKIPPED FOR TRAINING

| Model Name | Parameters | Accuracy | |
|---|---|---|---|
| | | ESC-50 | Speech Commands V2 |
| ENV2 | 8.74M | 86.74% (±0.29) | 97.10% (±0.06) |
| TFE(4)-ENV2 | 8.82M | **87.13%** (±0.43) | **97.19%** (±0.09) |

We also explored the impact of knowledge distillation on training models. In the experiment, by fixing the balancing coefficient $\lambda = 1$, the CMKD method was not performed to train the original ENV2 and the proposed TFE(4)-ENV2. Table IV shows the performance of ENV2 and TFE(4)-ENV2 when the CMKD was skipped. According to the results, the proposed TFE(4)-ENV2 still had better performance than the original ENV2 on both two datasets. However, compared to training with knowledge distillation, the accuracy of both ENV2 and TFE(4)-ENV2 dropped. For datasets ESC-50 and Speech Commands V2, the accuracy of the original ENV2 dropped from 88.63% to 86.74% and from 97.74% to 97.10%, respectively; the accuracy of the proposed TFE(4)-ENV2 dropped from 89.67% to 87.13% and from 97.85% to 97.19%, respectively. These findings demonstrate the importance of knowledge distillation for training the image-based model in the audio classification, and they also confirm that the proposed TFE module can still help improve the performance of the original ENV2 for speech classification even when knowledge distillation is not performed for training.

## V. CONCLUSIONS

Unlike the general field of image processing, where the vertical and horizontal axes typically have similar physical meanings, the audio spectrogram exhibits distinct interpretations for these axes. This unique characteristic poses a challenge when utilizing spectrograms for image-based classifier models. To address this issue and obtain a more suitable spectrogram for the image-based classifier model, we proposed a new TFE module for spectrogram enhancement. The proposed model, TFE, consists of multiple Enhancer Blocks to extract time-frequency features in high dimensions, providing enhanced spectrograms to ENV2 for classification. In our experiments, compared with the original ENV2, the accuracy of the proposed TFE-ENV2 increased by 1.04% and 0.11% on ESC-50 and Speech Commands V2 respectively. All of these results demonstrated the effectiveness of the proposed TFE for ENV2 on audio classification tasks. Moreover, the evidence supports the idea that enhancing spectrograms through the extraction of high-dimensional time-frequency features based on energy distributions can be beneficial for audio classification. These findings not only contribute to the current understanding of audio classification but also provide a potential direction for future research in this area.

## ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 23H03423.

## REFERENCES

[1] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Trans. Speech & Audio Process*, vol. 10, no. 7, pp. 504–516, 2002.

[2] J. Benesty, S. Makino, and J. Chen, *Speech enhancement*. Springer Science & Business Media, 2006.

[3] B. McMahan and D. Rao, "Listening to the world improves speech command recognition," in *arXiv preprint arXiv:1710.08377v1*, 2018.

[4] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? Adversarial examples against automatic speech recognition," *arXiv preprint arXiv:1801.00554*, 2018.

[5] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," vol. 3361, no. 10, pp. 255–258, 1998.

[6] Hershey *et al.*, "CNN architectures for large-scale audio classification," in *Proc. ICASSP*, 2017, pp. 131–135.

[7] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal processing letters*, vol. 24, no. 3, pp. 279–283, 2017.

[8] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio spectrogram transformer," *arXiv preprint arXiv:2104.01778*, 2021.

[9] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *arXiv preprint arXiv:2110.05069*, 2021.

[10] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," in *Proc. ICASSP*, 2022, pp. 646–650.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] G. Aguilar, Y. Ling, Y. Zhang, B. Yao, X. Fan, and C. Guo, "Knowledge distillation from internal representations," in *Proc. AAAI*, 2020, pp. 7350–7357.

[13] Y. Gong, S. Khurana, A. Rouditchenko, and J. Glass, "CMKD: CNN/Transformer-based cross-model knowledge distillation for audio classification," *arXiv preprint arXiv:2203.06760*, 2022.

[14] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, 2019, pp. 6105–6114.

[15] 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 preprint arXiv:1704.04861*, 2017.

[16] M. Tan and Q. Le, "EfficientNetV2: Smaller models and faster training," in *Proc. ICML*, 2021, pp. 10 096–10 106.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image

database," in *Proc. CVPR*, 2009, pp. 248–255.
[18] S. Gupta and M. Tan, "EfficientNet-EdgeTPU: Creating accelerator-optimized neural networks with automl," *Google AI Blog*, vol. 2, no. 1, 2019.
[19] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. ICML*, 2019, pp. 7354–7363.
[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. CVRP*, 2018, pp. 4510–4520.
[21] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVRP*, 2018, pp. 7132–7141.
[22] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, 2017, pp. 776–780.
[23] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.
[24] N. Zeghidour, O. Teboul, F. d. C. Quitry, and M. Tagliasacchi, "LEAF: A learnable frontend for audio classification," *arXiv preprint arXiv:2101.08596*, 2021.
[25] Q. Zhang, Q. Song, Z. Ni, A. Nicolson, and H. Li, "Time-frequency attention for monaural speech enhancement," in *Proc. ICASSP*, 2022, pp. 7852–7856.
[26] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. ACM*, 2015, pp. 1015–1018.
[27] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.
[28] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.
[29] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *arXiv preprint arXiv:1906.02629v3*, 2019.
[30] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," *arXiv preprint arXiv:1711.10282*, 2017.