INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

TFECN: Time-Frequency Enhanced ConvNet for Audio Classification

Mengwei Wang[1,2], *Zhe Yang*[1,2,*]

[1]School of Computer Science and Technology, Soochow University, Suzhou, China
[2]Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, China

mwwang@stu.suda.edu.cn, yangzhe@suda.edu.cn

Abstract

Recently, transformer-based models have shown leading performance in audio classification, gradually replacing the dominant ConvNet in the past. However, some research has shown that certain characteristics and designs in transformers can be applied to other architectures and make them achieve similar performance as transformers. In this paper, we introduce TFECN, a pure ConvNet that combines the design in transformers and has time-frequency enhanced convolution with large kernels. It can provide a global receptive field on the frequency dimension as well as avoid the influence of the convolution's shift-equivariance on the recognition of not shift-invariant patterns along the frequency axis. Furthermore, to use ImageNet-pretrained weights, we propose a method for transferring weights between kernels of different sizes. On the commonly used datasets AudioSet, FSD50K, and ESC50, our TFECN outperforms the models trained in the same way.

Index Terms: audio classification, large kernel ConvNet, transfer learning

1. Introduction

Audio classification, which refers to mapping an audio segment into one or more sound event categories, is an active research topic in acoustic signal processing. For the past few years, convolutional neural networks (ConvNets) have been the mainstream model for this task [1, 2, 3], but recently, their dominance has been significantly challenged by transformer-based models. Transformers were first used in natural language processing [4], and later, Vision Transformers [5, 6, 7] introduced them to computer vision and became a hot research topic. Starting with the Audio Spectrogram Transformer [8], an increasing number of transformer-based models for audio classification have emerged and continue to set new records on various datasets [9, 10]. As the knowledge of transformers has matured, researchers have begun to examine certain characteristics or designs of transformers to analyze their superior performance. Inspired by the global receptive field of the self-attention mechanism, RepLKNet [11] increased the receptive field of ConvNets by introducing very large kernels. Mlp-Mixer [12] and Metaformer [13] adopted the token-mixer and channel-mixer design of transformers and replaced the self-attention mechanism with an MLP or even an extremely simple pooling layer, empirically demonstrating that the framework of transformers plays an important role in their performance. ConvNext [14] analyzed the design space of transformers in detail and gradually introduced the designs into a traditional ResNet [15] so that its performance gradually approached and exceeded transformers. While these works summarized a variety of methods to improve model performance using the design of transformers, no work has yet attempted to use these methods to improve a ConvNet for audio classification, although ConvNet tends to have fewer parameters, less computation, and a simpler structural design compared to a transformer.

Unlike an image, the two axes of a spectrogram represent the individual frequency components and the time frames. Patterns aligned with the time axis are shift-invariant, similar to the objects in an image, which means that the shift of a visual pattern along the time axis can be seen as a change in the spatial position of an object in the image. In contrast, patterns distributed along the frequency axis are not shift-invariant [16], which means that if the visual patterns of a sound event category shifted along the frequency axis, the category or semantic represented by the patterns is likely to have changed, as shown in Figure 1(a). HTS-AT [9] is an audio classification model based on the Swin-Transformer [6]. It arranged spectrogram patches in a time-frequency-window order to focus patches with different frequency components of the same time frame into a single attention window, making the model predict only along the time axis. Although the sliding window that shifts along both axes in the 2D conv can naturally fit the shift-invariant patterns distributed along the time axis in the spectrogram, it also conflicts with the not shift-invariant patterns on the frequency dimension. MMDenseNet [17] divided the frequency dimension of the spectrogram into multiple frequency bands and convolved them separately using different kernels. This approach avoided recognizing the same visual patterns as having the same semantics along the entire frequency axis with the same weights but at the expense of the receptive field on the frequency dimension. These works inspired us to propose a new conv method to better fit spectrogram features.

Currently, for audio classification, the best models are almost all transformer-based models. To demonstrate that a pure ConvNet can still achieve leading performance, we introduce the time-frequency enhanced ConvNet (TFECN), which combines the generic architecture design in transformers and improves the convolution for spectrogram features. The main contributions of our work are as follows:

- TFECN outperforms models trained in the same way (ImageNet pretraining, then supervised training on AudioSet, and fine-tuning for downstream tasks) on three datasets, AudioSet [18], FSD50K [19], and ESC50 [20], demonstrating that a pure ConvNet can still achieve advanced performance in audio classification.
- Our proposed time-frequency enhanced convolution, not only releases the shift-equivariance of convolution on the frequency dimension but also provides the global receptive field on the frequency dimension.

* corresponding author

10.21437/Interspeech.2023-734

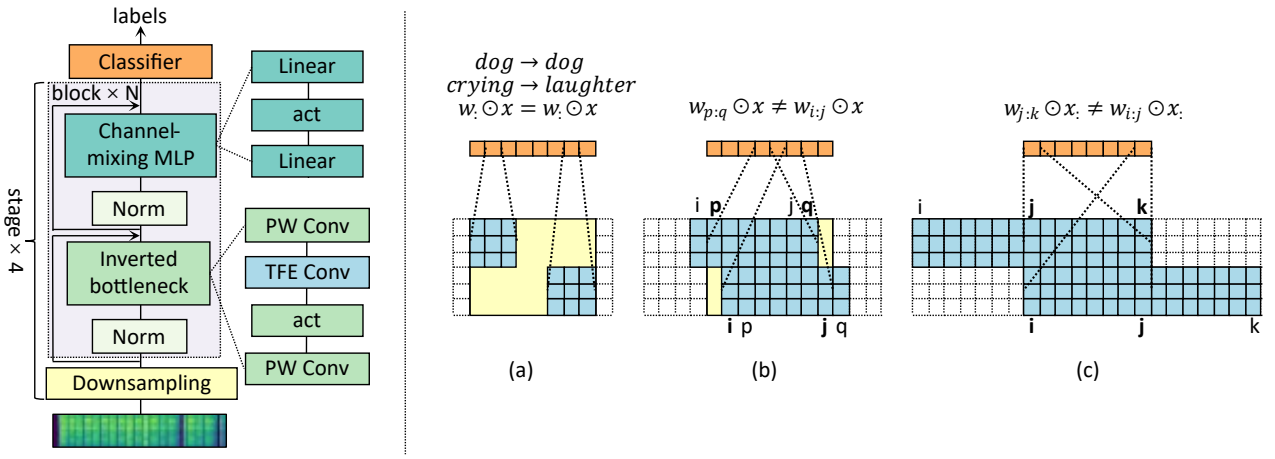

Figure 1: *Left shows the overall architecture of TFECN. For the Norm layer, we use Layer Normalization [21] rather than Batch Norm [22], which has shown effectiveness in recent research [14]. We use Squared ReLu [23] as the activation function, similar to the pretrained model [24]. Right is an illustration of the FEConv. For convenience, we assume that all regions covered by conv kernels (blue) in the input feature map ($X \in \mathbb{R}^{T \times F}$, yellow) have the same value and represent the output feature map (orange) as a one-dimensional form that retains only the frequency dimension.*

- To use the pretrained weights provided by previous work, we propose a method to transfer weights from small kernels to larger kernels. This method allows models with large kernels to utilize models with similar structures to them but using smaller kernels for transfer learning.

2. Time-frequency enhanced ConvNet

2.1. Model architecture

The overall architecture of TFECN is shown on the left in Figure 1. Our model is based on the common four-stage design of transformers with a stage compute ratio of 3:3:9:3. The different stages are separated by downsampling layers using a 3×3 conv with stride 2. At the beginning of the model, a patchify layer consisting of 7×7 conv with stride 4 is used to generate the spectrogram patches. Inside the basic blocks that make up the stage, there is first a conv block that mainly recognizes spatial patterns, which uses the inverted bottleneck design [25] with an expansion ratio of 2. In the inverted bottleneck, the feature dimension is first expanded by a pointwise conv with 1×1 kernels, then the spatial patterns are captured by a depthwise conv in which the number of groups is equal to the number of channels, and finally, the feature dimension is recovered by another pointwise conv.

In the depthwise separable conv, which consists of pointwise conv and depthwise conv, the pointwise conv is responsible for capturing cross-channel patterns. In the inverted bottleneck, we believe that pointwise conv is more likely to apply many different projections of feature maps into different feature subspaces, as in the case of the Q, K, V, and head transformations in the transformers' self-attention mechanism. In this way, the inter-channel communication of the model may be reduced. Transformers added an MLP after the self-attention layer as a complement to improve this problem. Inspired by this, we added an additional MLP after the inverted bottleneck to increase the communication between the different feature dimensions. In addition, for more efficient training, we set the learnable scale and bias after the activation function and scaled the shortcuts in the last two stages [24].

2.2. Time-frequency enhanced convolution

Since patterns distributed along the frequency axis in the spectrogram are not shift-invariant, a kernel smaller than the feature map sliding within the feature map can interfere with recognition, especially for the top layer of the model that recognizes high-level semantic features. As shown in Figure 1(a), the same visual patterns at different positions in the feature map are connected to the neurons through the same weights, which eventually produce the same output:

$$w_: \odot x = w_: \odot x \tag{1}$$

where $w_:$ denotes all columns in the kernel. This means that the pattern at different positions has the same semantics. In the image, this corresponds to reality, for example, a dog moving from left to right. But when a time-frequency pattern moves from low to high frequencies in a spectrogram, the semantics may change, for example, from crying to laughter. To prevent the kernel from recognizing the same visual patterns as the same semantics along the frequency axis, we first expand the kernel along the frequency direction until it covers the whole frequency dimension so that the weights connecting the neurons to the feature maps are always different when the kernel slides along the frequency axis. As shown in Figure 1(b), even two identical visual patterns connected to neurons with different weights will eventually produce different outputs, and different outputs mean different semantics, which can be expressed as

$$w_{p:q} \odot x \neq w_{i:j} \odot x \tag{2}$$

where $w_{p:q}$ denotes columns p to q in the kernel. Finally, to ensure that each neuron has a receptive field covering the entire frequency dimension, we continue to expand the kernel along the frequency axis until Equation 3 is satisfied.

$$K_f - \left\lfloor \frac{K_f}{2} \right\rfloor = F \tag{3}$$

For an input feature map $X \in \mathbb{R}^{T \times F}$, F denotes its size along the frequency direction. K_f denotes the size of the conv kernel

along the frequency direction. In this way, after padding the input feature map so that its size does not change before and after conv, it is still guaranteed that the leftmost and rightmost neurons in the output feature map can cover the entire frequency dimension, as shown in Figure 1(c). We call the depthwise conv using such kernels the frequency enhanced convolution (FEConv).

Considering that temporal cues also play an important role in recognizing some sound events, we introduce another kernel and extend it along the time direction to increase the receptive field of the time dimension so that it can better recognize the shift-invariant features on the time dimension. We call the depthwise conv using such kernels the time enhanced convolution (TEConv). Finally, we use the time-frequency enhanced convolution (TFEConv) consisting of TEConv and FEConv together to replace the depthwise conv in the inverted bottleneck. In TFEConv, we sum the weighted outputs of TEConv and FEConv using two learnable weight parameters w_t and w_f, which can be expressed as Equation 4.

$$TFEConv(X)=w_t\cdot(TEConv(X))+w_f\cdot(FEConv(X)) \tag{4}$$

2.3. ImageNet pretraining

We perform transfer learning using the ImageNet-pretrained model, as in previous works [2, 8, 9, 10], and using convformer_s18_384_in21ft1k [1] as the pretrained model. However, this pretrained model has kernel sizes of 7×7 and is not directly transferable to our model. Inspired by the method of merging kernels of different sizes in structural re-parameterization [26, 27], we first initialize two large kernels (LKs, LK $\in \mathbb{R}^{K_t \times K_f}$) of the same size as in TEConv and FEConv and later assign weights to the central regions of the two large kernels using the weights of the pretrained small kernels (PSKs, PSK$\in \mathbb{R}^{K_s \times K_s}$ and $K_s \leq min(K_t, K_f)$). We call the kernels formed in this way the pretrained large kernels (PLKs, PLK $\in \mathbb{R}^{K_t \times K_f}$). Because of the additivity of convolution, the conv using PLKs can be expressed as

$$X \otimes PLK = X \otimes LK + X \otimes PSK \tag{5}$$

where $X \in \mathbb{R}^{T \times F}$ denotes the input and $\otimes$ denotes the convolution operation. To avoid undermining the knowledge learned from pretraining, we use zero to initialize LKs. Thus, at the beginning of training,

$$X \otimes PLK = X \otimes PSK \tag{6}$$

Due to the pretrained weights, weights with a value of zero in PLKs can also be updated. Finally, the weights of PLKs are used to initialize the TFEConv weights.

In fact, transferring weights between kernels of the same size is equivalent to initializing a kernel of the same size as the pretrained kernel using zero and adding the pretrained kernel to it. Thus, our proposed weight transfer method simply replaces the kernel of the same size with a larger kernel and does not impair the transfer of the knowledge learned from pretraining between different kernels.

3. Experiments

3.1. Datasets

AudioSet [18] is the largest public dataset for audio classification, consisting of over 2 million 10-second audio clips extracted from videos on YouTube and manually labeled into 527 categories. The entire dataset is officially divided into a balanced training set, an unbalanced training set and an eval set. Considering that the videos on YouTube disappear over time and to ensure the comparability of our experimental results, we use the method provided by previous work [1] to download the dataset. Finally, we downloaded 1,912,137 clips from the unbalanced training set, 20,550 clips from the balanced training set, and 18,887 clips from the eval set. We use the full training set consisting of the unbalanced training set and the balanced training set to train our model and evaluate our model on the officially divided eval set.

FSD50K [19] is the second largest public dataset of human-labeled sound events containing 51K audio clips picked from Freesound with an average duration of 7.6 s and manually labeled into 200 sound event categories drawn from the AudioSet ontology. We use the officially split training set, validation set, and eval set. Considering the variable clip lengths in FSD50K, we first unify all the clip lengths to 10 s. For clips longer than 10 s, we divide the original clips into multiple samples with a certain overlap. Specifically, we use a 3 s overlap to segment the clips in the training set, while the validation and eval sets are segmented without using the overlap. In addition, FSD50K's carefully curated evaluation set reduces the impact of noise on the metrics that reflect model performance, more accurately reflecting the differences in performance between models. For this reason, we chose to conduct ablation experiments on FSD50K.

ESC50 [20] contains 2000 5-second audio clips and is manually labeled using 50 categories. we repeat each clip twice to be consistent with the 10-second length in the other two datasets and perform 5-fold cross-validation using the official division of 5 folds.

For AudioSet and FSD50K, we use mean Average Precision (mAP) as the metric, and for ESC50 we use Accuracy (Acc) as the metric. All metrics are chosen to facilitate comparison with existing methods.

3.2. Implementation details

First, we convert each clip to monophonic, resample to 32 kHz, and unify to 10 s. Then, a Hann window of size 1024 and a hop size of 320 are used to compute STFTs, and the frequency range is limited to between 50 Hz and 14 kHz. Finally, 128 Mel filter banks are used to compute the log Mel spectrograms, resulting in spectrogram features with the shape (1001, 128).

We denote the size of the TFEConv in a stage as (K_t, K_f, K_s), where K_t is the size along the time axis in TEConv, K_f is the size along the frequency axes in FEConv, and K_s is the size of the remaining directions in the two kernels. After patchify stem and downsampling, the sizes of the features along the frequency axis in the four stages are 32, 16, 8, and 4. According to Equation 3, we set K_f to 31, 15, and 7 in TFEConv for the last three stages. Due to the weaker conv layer in the first stage and the larger feature map size, we set K_f to 31 so that it covers almost the entire frequency dimension, releasing the shift-equivariance while reducing FLOPs. Finally, we set the kernel sizes for the four stages to [(15, 31, 7), (15, 31, 7), (15, 15, 7), (15, 7, 7)] and use the efficient implementation of large kernel depthwise conv in RepLKNet [11]. In addition, to avoid corrupting the learned knowledge when using pretrained weights, we set the defaults of w_f and w_t in TFEConv to 0.5 in all four stages.

Mix-up [28] with α=0.5, SpecAug [29] with a maximum

[1] https://github.com/sail-sg/metaformer

Table 1: *The results of ablation experiments on FSD50K*

	Params	MACs	val mAP	eval mAP
MobileNetV2-like	11.0M	4.0G	.529	.497
DSCN	25.1M	10.1G	.549	.519
TFECN	26.7M	11.9G	.557	.515
DSCN-ImgP	25.1M	10.1G	.622	.598
TFECN-ImgP	26.7M	11.9G	.625	.601

of 256 time-mask, 64 frequency-mask, and label smoothing with smoothing=0.1 are used for data augmentation. All models are trained using the AdamW optimizer (β_1=0.9, β_2=0.999, eps=1e-8, decay=0.05). After 10 epochs of warm-up, the learning rate first grows linearly from 0 to 4e-4 and then decays to 1e-7 after 40 epochs using the cosine annealing scheduler. For ESC50, the training continues for a total of 100 epochs with a learning rate of 1e-7. We also use the exponential moving average (EMA) optimization strategy to improve the generality of the models. For different classifications, binary cross-entropy is used as the loss for AudioSet and FSD50K, and cross-entropy is used for ESC50. For AudioSet, we use the weighted sampler [2, 10], to sample 200K samples from the full training set in each epoch and train on 4 NVIDIA A-100 GPUs with a batch size of 128 for approximately 10 hours. For FSD50K and ESC50, use the entire training data as the training set for each epoch and train on 2 NVIDIA A-100 GPUs with batch sizes of 64 and 16, respectively.

3.3. Results

3.3.1. Ablation experiments

The ablation experiments start with a MobileNetV2-like model, which has two simplifications compared to TFECN: removing the MLP and replacing TFEConv with depthwise conv with 7×7 kernels. Then, we reintroduce the MLP. Since MLP is equivalent to pointwise convolution in depthwise separable conv and inverted bottleneck is equivalent to depthwise conv, we call this model depthwise separable ConvNet (DSCN). Next, we use TFEConv to replace the depthwise conv in the inverted bottleneck to form the TFECN. Furthermore, to verify the validity of the proposed weight transfer method, we use ImageNet-pretrained weights for transferring weights between kernels of the same size (DSCN-ImgP) and between kernels of different sizes (TFECN-ImgP). As shown in Table 1, the performance of MobileNetV2-like was improved by introducing MLP, but overfitting was observed in TFECN. The possible reason is the loss of the locality prior when training large kernels individually, leading to difficulties in optimizing on small datasets and causing a loss of generality [11]. Similar to the small kernels used by RepLKNet [11] to assist in training the large kernel models, the pretrained weights of the small kernels we introduce in the large kernels also play the role of reintroducing the locality prior. Therefore, overfitting is mitigated and eventually makes TFECN-ImgP better than DSCN-ImgP. In summary, while TFEConv can eventually improve the model, it has a significant dependence on pretrained weights.

3.3.2. AudioSet experiments

In Table 2, we compare TFECN with several of the latest models in audio classification. Among these methods, except for PANN and ERANN, which are trained from scratch, the rest of the models all use ImageNet-pretrained weights. The results show that TFECN significantly outperforms the previous ConvNets and outperforms the latest Transformer-based models with fewer parameters.

Table 2: *Performance comparison of TFECN and previous methods on AudioSet.*

	Architecture	Params	Pretraining	mAP
PANN [1]	ConvNet	81M	-	.431
PSLA [2]	ConvNet	13.6M	ImageNet	.444
ERANN [3]	ConvNet	55M	-	.450
AST [8]	transformer	86M	ImageNet	.459
HTS-AT [9]	transformer	31M	ImageNet	.471
PaSST [10]	transformer	86M	ImageNet	.471
TFECN	ConvNet	27M	ImageNet	.477

3.3.3. FSD50K and ESC50 experiments

As in previous work, we used AudioSet-pretrained TFECN to fine-tune on FSD50K and ESC50. The results are shown in Table 3. TFECN achieves new state-of-the-art results on FSD50K, outperforming the previous best transformer-based model. On ESC50, TFECN also outperforms the models trained in the same way.

Table 3: *Performance comparison of TFECN and previous methods on FSD50K and ESC50.*

	Pretraining	FSD50K mAP	ESC50 Acc
PANN [1]	Audio	-	94.7
PSLA [2]	Img+Audio	56.71	-
ERANN [3]	-	-	96.1
AST [8]	Img+Audio	-	95.7
HTS-AT [9]	Img+Audio	-	97.0
PaSST [10]	Img+Audio	65.55	-
TFECN	Img+Audio	67.29	97.7

4. Conclusions

In this paper, we introduce TFECN, a pure ConvNet for audio classification, which achieves outstanding performance on several datasets, outperforming recent transformer-based models, demonstrating that a pure ConvNet can still achieve advanced performance in audio classification. However, TFECN is overly dependent on pretraining due to the difficulty of training large kernels. In the future, we will continue to explore efficient ways to train TFECN so that it can still achieve excellent performance without pretraining.

5. Acknowledgements

This work was supported by the National Natural Science Foundation of China (61772356), the Project of the Ministry of Education on the Cooperation of Production and Education (220606363154256), the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

6. References

[1] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[2] Y. Gong, Y.-A. Chung, and J. Glass, "Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3292–3306, 2021.

[3] S. Verbitskiy, V. Berikov, and V. Vyshegorodtsev, "Eranns: Efficient residual audio neural networks for audio pattern recognition," *Pattern Recognition Letters*, vol. 161, pp. 38–44, 2022.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[6] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 012–10 022.

[7] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, 18–24 Jul 2021, pp. 10 347–10 357.

[8] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," in *Proc. Interspeech 2021*, 2021, pp. 571–575.

[9] 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 *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 646–650.

[10] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Proc. Interspeech 2022*, 2022, pp. 2753–2757.

[11] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31x31: Revisiting large kernel design in cnns," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 963–11 975.

[12] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, "Mlp-mixer: An all-mlp architecture for vision," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 24 261–24 272.

[13] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 819–10 829.

[14] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 976–11 986.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[16] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, "Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection," in *Proc. Interspeech 2022*, 2022, pp. 2763–2767.

[17] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 21–25.

[18] J. F. Gemmeke, D. P. W. 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[19] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: An open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2022.

[20] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, p. 1015–1018.

[21] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[22] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[23] D. R. So, W. Mańke, H. Liu, Z. Dai, N. Shazeer, and Q. V. Le, "Primer: Searching for efficient transformers for language modeling," *arXiv preprint arXiv:2109.08668*, 2021.

[24] W. Yu, C. Si, P. Zhou, M. Luo, Y. Zhou, J. Feng, S. Yan, and X. Wang, "Metaformer baselines for vision," *arXiv preprint arXiv:2210.13452*, 2022.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[26] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 13 733–13 742.

[27] X. Ding, X. Zhang, J. Han, and G. Ding, "Diverse branch block: Building a convolution as an inception-like unit," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 10 886–10 895.

[28] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[29] 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," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.