INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

AST: Audio Spectrogram Transformer

Yuan Gong, Yu-An Chung, James Glass

MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
{yuangong, andyyuan, glass}@mit.edu

Abstract

In the past decade, convolutional neural networks (CNNs) have been widely adopted as the main building block for end-to-end audio classification models, which aim to learn a direct mapping from audio spectrograms to corresponding labels. To better capture long-range global context, a recent trend is to add a self-attention mechanism on top of the CNN, forming a CNN-attention hybrid model. However, it is unclear whether the reliance on a CNN is necessary, and if neural networks purely based on attention are sufficient to obtain good performance in audio classification. In this paper, we answer the question by introducing the *Audio Spectrogram Transformer* (AST), the first convolution-free, purely attention-based model for audio classification. We evaluate AST on various audio classification benchmarks, where it achieves new state-of-the-art results of 0.485 mAP on AudioSet, 95.6% accuracy on ESC-50, and 98.1% accuracy on Speech Commands V2.
Index Terms: audio classification, self-attention, Transformer

1. Introduction

With the advent of deep neural networks, over the last decade audio classification research has moved from models based on hand-crafted features [1, 2] to end-to-end models that directly map audio spectrograms to corresponding labels [3, 4, 5]. Specifically, convolutional neural networks (CNNs) [6] have been widely used to learn representations from raw spectrograms for end-to-end modeling, as the inductive biases inherent to CNNs such as spatial locality and translation equivariance are believed to be helpful. In order to better capture long-range global context, a recent trend is to add a self-attention mechanism on top of the CNN. Such CNN-attention hybrid models have achieved state-of-the-art (SOTA) results for many audio classification tasks such as audio event classification [7, 8], speech command recognition [9], and emotion recognition [10]. However, motivated by the success of purely attention-based models in the vision domain [11, 12, 13], it is reasonable to ask whether a CNN is still essential for audio classification.

To answer the question, we introduce the *Audio Spectrogram Transformer* (AST), a *convolution-free*, *purely* attention-based model that is directly applied to an audio spectrogram and can capture long-range global context even in the lowest layers. Additionally, we propose an approach for transferring knowledge from the Vision Transformer (ViT) [12] pretrained on ImageNet [14] to AST, which can significantly improve the performance. The advantages of AST are threefold. First, AST has superior performance: we evaluate AST on a variety of audio classification tasks and datasets including AudioSet [15], ESC-50 [16] and Speech Commands [17]. AST outperforms state-of-the-art systems on all these datasets. Second, AST naturally supports variable-length inputs and can be applied to different tasks without any change of architecture. Specifically, the models we use for all aforementioned tasks have the same architecture while the input lengths vary from 1 sec. (Speech Commands) to 10 sec. (AudioSet). In contrast, CNN-based models typically require architecture tuning to obtain optimal performance for different tasks. Third, comparing with SOTA CNN-attention hybrid models, AST features a simpler architecture with fewer parameters, and converges faster during training. To the best of our knowledge, AST is the first purely attention-based audio classification model.

Code at https://github.com/YuanGongND/ast.

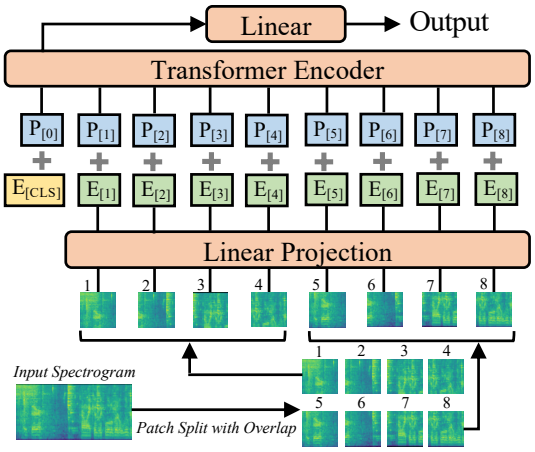

Figure 1: *The proposed audio spectrogram transformer (AST) architecture. The 2D audio spectrogram is split into a sequence of 16×16 patches with overlap, and then linearly projected to a sequence of 1-D patch embeddings. Each patch embedding is added with a learnable positional embedding. An additional classification token is prepended to the sequence. The output embedding is input to a Transformer, and the output of the classification token is used for classification with a linear layer.*

Related Work The proposed Audio Spectrogram Transformer, as the name suggests, is based on the Transformer architecture [18], which was originally proposed for natural language processing tasks. Recently, the Transformer has also been adapted for audio processing, but is typically used in conjunction with a CNN [19, 20, 21]. In [19, 20], the authors stack a Transformer on top of a CNN, while in [21], the authors combine a Transformer and a CNN in each model block. Other efforts combine CNNs with simpler attention modules [8, 7, 9]. The proposed AST differs from these studies in that it is convolution-free and purely based on attention mechanisms. The closest work to ours is the Vision Transformer (ViT) [11, 12, 13], which is a Transformer architecture for vision tasks. AST and ViT have similar architectures but ViT has only been applied to fixed-dimensional inputs (images) while AST can process variable-length audio inputs. In addition, we propose an approach to transfer knowledge from ImageNet pretrained ViT to AST. We also conduct extensive experiments to show the design choice of AST on audio tasks.

http://dx.doi.org/10.21437/Interspeech.2021-698

2. Audio Spectrogram Transformer

2.1. Model Architecture

Figure 1 illustrates the proposed Audio Spectrogram Transformer (AST) architecture. First, the input audio waveform of t seconds is converted into a sequence of 128-dimensional log Mel filterbank (fbank) features computed with a 25ms Hamming window every 10ms. This results in a $128 \times 100t$ spectrogram as input to the AST. We then split the spectrogram into a sequence of N 16×16 patches with an overlap of 6 in both time and frequency dimension, where $N = 12\lceil(100t - 16)/10\rceil$ is the number of patches and the effective input sequence length for the Transformer. We flatten each 16×16 patch to a 1D patch embedding of size 768 using a linear projection layer. We refer to this linear projection layer as the patch embedding layer. Since the Transformer architecture does not capture the input order information and the patch sequence is also not in temporal order, we add a trainable positional embedding (also of size 768) to each patch embedding to allow the model to capture the spatial structure of the 2D audio spectrogram.

Similar to [22], we append a `[CLS]` token at the beginning of the sequence. The resulting sequence is then input to the Transformer. A Transformer consists of several encoder and decoder layers. Since AST is designed for classification tasks, we only use the encoder of the Transformer. Intentionally, we use the original Transformer encoder [18] architecture without modification. The advantages of this simple setup are 1) the standard Transformer architecture is easy to implement and reproduce as it is off-the-shelf in TensorFlow and PyTorch, and 2) we intend to apply transfer learning for AST, and a standard architecture makes transfer learning easier. Specifically, the Transformer encoder we use has an embedding dimension of 768, 12 layers, and 12 heads, which are the same as those in [12, 11]. The Transformer encoder's output of the `[CLS]` token serves as the audio spectrogram representation. A linear layer with sigmoid activation maps the audio spectrogram representation to labels for classification.

Strictly speaking, the patch embedding layer can be viewed as a single convolution layer with a large kernel and stride size, and the projection layer in each Transformer block is equivalent to 1×1 convolution. However, the design is different from conventional CNNs that have multiple layers and small kernel and stride sizes. These Transformer models are usually referred to as convolution-free to distinguish them from CNNs [11, 12].

2.2. ImageNet Pretraining

One disadvantage of the Transformer compared with CNNs is that the Transformer needs more data to train [11]. In [11], the authors point out that the Transformer only starts to outperform CNNs when the amount of data is over 14 million for image classification tasks. However, audio datasets typically do not have such large amounts of data, which motivates us to apply cross-modality transfer learning to AST since images and audio spectrograms have similar formats. Transfer learning from vision tasks to audio tasks has been previously studied in [23, 24, 25, 8], but only for CNN-based models, where ImageNet-pretrained CNN weights are used as initial CNN weights for audio classification training. In practice, it is computationally expensive to train a state-of-the-art vision model, but many commonly used architectures (e.g., ResNet [26], EfficientNet [27]) have off-the-shelf ImageNet-pretrained models for both TensorFlow and PyTorch, making transfer learning much easier. We also follow this regime by adapting an off-the-shelf pretrained Vision Transformer (ViT) to AST.

While ViT and AST have similar architectures (e.g., both use a standard Transformer, same patch size, same embedding size), they are not same. Therefore, a few modifications need to make for the adaptation. First, the input of ViT is a 3-channel image while the input to the AST is a single-channel spectrogram, we average the weights corresponding to each of the three input channels of the ViT patch embedding layer and use them as the weights of the AST patch embedding layer. This is equivalent to expanding a single-channel spectrogram to 3-channels with the same content, but is computationally more efficient. We also normalize the input audio spectrogram so that the dataset mean and standard deviation are 0 and 0.5, respectively. Second, the input shape of ViT is fixed (either 224×224 or 384×384), which is different from a typical audio spectrogram. In addition, the length of an audio spectrogram can be variable. While the Transformer naturally supports variable input length and can be directly transferred from ViT to AST, the positional embedding needs to be carefully processed because it learns to encode the spatial information during the ImageNet training. We propose a cut and bi-linear interpolate method for positional embedding adaptation. For example, for a ViT that takes 384×384 image input and uses a patch size of 16×16, the number of patches and corresponding positional embedding is $24 \times 24 = 576$ (ViT splits patches without overlap). An AST that takes 10-second audio input has 12×100 patches, each patch needs a positional embedding. We therefore cut the first dimension and interpolate the second dimension of the 24×24 ViT positional embedding to 12×100 and use it as the positional embedding for the AST. We directly reuse the positional embedding for the `[CLS]` token. By doing this we are able to transfer the 2D spatial knowledge from a pretrained ViT to the AST even when the input shapes are different. Finally, since the classification task is essentially different, we abandon the last classification layer of the ViT and reinitialize a new one for AST. With this adaptation framework, the AST can use various pretrained ViT weights for initialization. In this work, we use pretrained weights of a data-efficient image Transformer (DeiT) [12], which is trained with CNN knowledge distillation, 384×384 images, has 87M parameters, and achieves 85.2% top-1 accuracy on ImageNet 2012. During ImageNet training, DeiT has two `[CLS]` tokens; we average them as a single `[CLS]` token for audio training.

3. Experiments

In this section, we focus on evaluating the AST on AudioSet (Section 3.1) as weakly-labeled audio event classification is one of the most challenging audio classification tasks. We present our primary AudioSet results and ablation study in Section 3.1.2 and Section 3.1.3, respectively. We then present our experiments on ESC-50 and Speech Commands V2 in Section 3.2.

3.1. AudioSet Experiments

3.1.1. Dataset and Training Details

AudioSet [15] is a collection of over 2 million 10-second audio clips excised from YouTube videos and labeled with the sounds that the clip contains from a set of 527 labels. The balanced training, full training, and evaluation set contains 22k, 2M, and 20k samples, respectively. For AudioSet experiments, we use the exact same training pipeline with [8]. Specifically, we use ImageNet pretraining (as described in Section 2.2), balanced sampling (for full set experiments only), data augmenta-

Table 1: *Performance comparison of AST and previous methods on AudioSet.*

	Model Architecture	Balanced mAP	Full mAP
Baseline [15]	CNN+MLP	-	0.314
PANNs [7]	CNN+Attention	0.278	0.439
PSLA [8] (Single)	CNN+Attention	0.319	0.444
PSLA (Ensemble-S)	CNN+Attention	0.345	0.464
PSLA (Ensemble-M)	CNN+Attention	0.362	0.474
AST (Single)	Pure Attention	0.347 $\pm$ 0.001	0.459 $\pm$ 0.000
AST (Ensemble-S)	Pure Attention	0.363	0.475
AST (Ensemble-M)	Pure Attention	**0.378**	**0.485**

tion (including mixup [28] with mixup ratio=0.5 and spectrogram masking [29] with max time mask length of 192 frames and max frequency mask length of 48 bins), and model aggregation (including weight averaging [30] and ensemble [31]). We train the model with a batch size of 12, the Adam optimizer [32], and use binary cross-entropy loss. We conduct experiments on the official balanced and full training set and evaluate on the AudioSet evaluation set. For balanced set experiments, we use an initial learning rate of 5e-5 and train the model for 25 epochs, the learning rate is cut into half every 5 epoch after the 10th epoch. For full set experiments, we use an initial learning rate of 1e-5 and train the model for 5 epochs, the learning rate is cut into half every epoch after the 2nd epoch. We use the mean average precision (mAP) as our main evaluation metric.

3.1.2. AudioSet Results

We repeat each experiment three times with the same setup but different random seeds and report the mean and standard deviation. When AST is trained with the full AudioSet, the mAP at the last epoch is 0.448$\pm$0.001. As in [8], we also use weight averaging [30] and ensemble [31] strategies to further improve the performance of AST. Specifically, for weight averaging, we average all weights of the model checkpoints from the first to the last epoch. The weight-averaged model achieves an mAP of 0.459$\pm$0.000, which is our best single model (weight averaging does not increase the model size). For ensemble, we evaluate two settings: 1) Ensemble-S: we run the experiment three times with the exact same setting, but with a different random seed. We then average the output of the last checkpoint model of each run. In this setting, the ensemble model achieves an mAP of 0.475; 2) Ensemble-M: we ensemble models trained with different settings, specifically, we ensemble the three models in Ensemble-S together with another three models trained with different patch split strategies (described in Section 3.1.3 and shown in Table 5). In this setting, the ensemble model achieves an mAP of 0.485, this is our best full model on AudioSet. As shown in Table 1, the proposed AST outperforms the previous best system in [8] in all settings. Note that we use the same training pipeline with [8] and [8] also use ImageNet pretraining, so it is a fair comparison. In addition, we use fewer models (6) for our best ensemble models than [8] (10). Finally, it is worth mentioning that AST training converges quickly; AST only needs 5 training epochs, while in [8], the CNN-attention hybrid model is trained for 30 epochs.

We also conduct experiments with the balanced AudioSet (about 1% of the full set) to evaluate the performance of AST when the training data volume is smaller. For weight averaging, we average all weights of the model checkpoints of the last 20 epochs. For Ensemble-S, we follow the same setting used for the full AudioSet experiment; for Ensemble-M, we include 11 models trained with different random seeds (Table 1), different pretrained weights (Table 3), different positional embedding interpolation (Table 4), and different patch split strategies (Table 5). The single, Ensemble-S, and Ensemble-M model achieve 0.347$\pm$0.001, 0.363, and 0.378, respectively, all outperform the previous best system. This demonstrates that AST can work better than CNN-attention hybrid models even when the training set is relatively small.

Table 2: *Performance impact due to ImageNet pretraining. "Used" denotes the setting used by our optimal AST model.*

	Balanced Set	Full Set
No Pretrain	0.148	0.366
ImageNet Pretrain (Used)	0.347	0.459

Table 3: *Performance of AST models initialized with different ViT weights on balanced AudioSet and corresponding ViT models' top-1 accuracy on ImageNet 2012. (* Model is trained without patch split overlap due to memory limitation.)*

	# Params	ImageNet	AudioSet
ViT Base [11]	86M	0.846	0.320
ViT Large [11]*	307M	0.851	0.330
DeiT w/o Distill [12]	86M	0.829	0.330
DeiT w/ Distill (Used)	87M	0.852	0.347

3.1.3. Ablation Study

We conduct a series of ablation studies to illustrate the design choices for the AST. To save compute, we mainly conduct ablation studies with the balanced AudioSet. For all experiments, we use weight averaging but do not use ensembles.

Impact of ImageNet Pretraining. We compare ImageNet pretrained AST and randomly initialized AST. As shown in Table 2, ImageNet pretrained AST noticeably outperforms randomly initialized AST for both balanced and full AudioSet experiments. The performance improvement of ImageNet pretraining is more significant when the training data volume is smaller, demonstrating that ImageNet pretraining can greatly reduce the demand for in-domain audio data for AST. We further study the impact of pretrained weights used. As shown in Table 3, we compare the performance of AST models initialized with pretrained weights of ViT-Base, ViT-Large, and DeiT models. These models have similar architectures but are trained with different settings. We made the necessary architecture modifications for AST to reuse the weights. We find that AST using the weights of the DeiT model with distillation that performs best on ImageNet2012 also performs best on AudioSet.

Impact of Positional Embedding Adaptation. As mentioned in Section 2.2, we use a cut and bi-linear interpolation approach for positional embedding adaptation when transferring knowledge from the Vision Transformer to the AST. We compare it with a pretrained AST model with a randomly initialized positional embedding. As shown in Table 4, we find reinitializing the positional embedding does not completely break the pretrained model as the model still performs better than a fully randomly reinitialized model, but it does lead to a noticeable performance drop compared with the proposed adaptation approach. This demonstrates the importance of transferring spatial

Table 4: *Performance impact due to various positional embedding adaptation settings.*

	Balanced Set
Reinitialize	0.305
Nearest Neighbor Interpolation	0.346
Bilinear Interpolation (Used)	0.347

Table 5: *Performance impact due to various patch overlap size.*

	# Patches	Balanced Set	Full Set
No Overlap	512	0.336	0.451
Overlap-2	657	0.342	0.456
Overlap-4	850	0.344	0.455
Overlap-6 (Used)	1212	0.347	0.459

Table 6: *Performance impact due to various patch shape and size. All models are trained with no patch split overlap.*

	# Patches	w/o Pretrain	w/ Pretrain
128×2	512	0.154	-
16×16 (Used)	512	0.143	0.336
32×32	128	0.139	-

knowledge. Bi-linear interpolation and nearest-neighbor interpolation do not result in a big difference.

Impact of Patch Split Overlap. We compare the performance of models trained with different patch split overlap [13]. As shown in Table 5, the performance improves with the overlap size for both balanced and full set experiments. However, increasing the overlap also leads to longer patch sequence inputs to the Transformer, which will quadratically increase the computational overhead. Even with no patch split overlap, AST can still outperform the previous best system in [8].

Impact of Patch Shape and Size. As mentioned in Section 2.1, we split the audio spectrogram into 16×16 square patches, so the input sequence to the Transformer cannot be in temporal order. We hope the positional embedding can learn to encode the 2D spatial information. An alternative way to split the patch is slicing the audio spectrogram into rectangular patches in the temporal order. We compare both methods in Table 6, when the area of the patch is the same (256), using 128×2 rectangle patches leads to better performance than using 16×16 square patches when both models are trained from scratch. However, considering there is no 128×2 patch based ImageNet pretrained models, using 16×16 patches is still the current optimal solution. We also compare using patches with different sizes, smaller size patches lead to better performance.

3.2. Results on ESC-50 and Speech Commands

The ESC-50 [16] dataset consists of 2,000 5-second environmental audio recordings organized into 50 classes. The current best results on ESC-50 are 86.5% accuracy (trained from scratch, SOTA-S) [33] and 94.7% accuracy (with AudioSet pretraining, SOTA-P) [7]. We compare AST with the SOTA models in these two settings, specifically, we train an AST model with only ImageNet pretraining (AST-S) and an AST model with ImageNet and AudioSet pretraining (AST-P). We train both models with frequency/time masking [29] data augmentation, a batch size of 48, and the Adam optimizer [32] for 20 epochs. We use an initial learning rate of 1e-4 and 1e-5 for AST-S and AST-P, respectively, and decrease the learning rate with a factor of 0.85 every epoch after the 5th epoch. We follow the standard 5-fold cross-validation to evaluate our model, repeat each experiment three times, and report the mean and standard deviation. As shown in Table 7, AST-S achieves 88.7±0.7 and AST-P achieves 95.6±0.4, both outperform SOTA models in the same setting. Of note, although ESC-50 has 1,600 training samples for each fold, AST still works well with such a small amount of data even without AudioSet pretraining.

Table 7: *Comparing AST and SOTA models on ESC-50 and Speech Commands. "-S" and "-P" denotes model trained without and with additional audio data, respectively.*

	ESC-50	Speech Commands V2 (35 classes)
SOTA-S	86.5 [33]	97.4 [34]
SOTA-P	94.7 [7]	97.7 [35]
AST-S	88.7±0.7	**98.11±0.05**
AST-P	**95.6±0.4**	97.88±0.03

Speech Commands V2 [17] is a dataset consists of 105,829 1-second recordings of 35 common speech commands. The training, validation, and test set contains 84,843, 9,981, and 11,005 samples, respectively. We focus on the 35-class classification task, the SOTA model on Speech Commands V2 (35-class classification) without additional audio data pretraining is the time-channel separable convolutional neural network [34], which achieves 97.4% on the test set. In [35], a CNN model pretrained with additional 200 million YouTube audio achieves 97.7% on the test set. We also evaluate AST in these two settings. Specifically, we train an AST model with only ImageNet pretraining (AST-S) and an AST model with ImageNet and AudioSet pretraining (AST-P). We train both models with frequency and time masking [29], random noise, and mixup [28] augmentation, a batch size of 128, and the Adam optimizer [32]. We use an initial learning rate of 2.5e-4 and decrease the learning rate with a factor of 0.85 every epoch after the 5th epoch. We train the model for up to 20 epochs, and select the best model using the validation set, and report the accuracy on the test set. We repeat each experiment three times and report the mean and standard deviation. AST-S model achieves 98.11±0.05, outperforms the SOTA model in [9]. In addition, we find AudioSet pretraining unnecessary for the speech command classification task as AST-S outperforms AST-P. To summarize, while the input audio length varies from 1 sec. (Speech Commands), 5 sec. (ESC-50) to 10 sec. (AudioSet) and content varies from speech (Speech Commands) to non-speech (AudioSet and ESC-50), we use a fixed AST architecture for all three benchmarks and achieve SOTA results on all of them. This indicates the potential for AST use as a generic audio classifier.

4. Conclusions

Over the last decade, CNNs have become a common model component for audio classification. In this work, we find CNNs are not indispensable, and introduce the Audio Spectrogram Transformer (AST), a convolution-free, purely attention-based model for audio classification which features a simple architecture and superior performance.

5. Acknowledgements

This work is partly supported by Signify.

6. References

[1] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in openSMILE, the Munich open-source multimedia feature extractor," in *Multimedia*, 2013.

[2] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, M. Mortillaro, H. Salamin, A. Polychroniou, F. Valente, and S. K. Kim, "The Interspeech 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism," in *Interspeech*, 2013.

[3] N. Jaitly and G. Hinton, "Learning a better representation of speech soundwaves using restricted boltzmann machines," in *ICASSP*, 2011.

[4] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *ICASSP*, 2014.

[5] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *ICASSP*, 2016.

[6] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, p. 1995, 1995.

[7] 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 TASLP*, vol. 28, pp. 2880–2894, 2020.

[8] Y. Gong, Y.-A. Chung, and J. Glass, "PSLA: Improving audio event classification with pretraining, sampling, labeling, and aggregation," *arXiv preprint arXiv:2102.01243*, 2021.

[9] O. Rybakov, N. Kononenko, N. Subrahmanya, M. Visontai, and S. Laurenzo, "Streaming keyword spotting on mobile devices," in *Interspeech*, 2020.

[10] P. Li, Y. Song, I. V. McLoughlin, W. Guo, and L.-R. Dai, "An attention pooling based representation learning method for speech emotion recognition," in *Interspeech*, 2018.

[11] 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 *ICLR*, 2021.

[12] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020.

[13] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," *arXiv preprint arXiv:2101.11986*, 2021.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *CVPR*, 2009.

[15] 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 *ICASSP*, 2017.

[16] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Multimedia*, 2015.

[17] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[19] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution augmented transformer for semi-supervised sound event detection," in *DCASE*, 2020.

[20] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound event detection of weakly labelled data with CNN-transformer and automatic threshold optimization," *IEEE/ACM TASLP*, vol. 28, pp. 2450–2460, 2020.

[21] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech*, 2020.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019.

[23] G. Gwardys and D. M. Grzywczak, "Deep image features in music information retrieval," *IJET*, vol. 60, no. 4, pp. 321–326, 2014.

[24] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "ESResNet: Environmental sound classification based on visual domain models," in *ICPR*, 2020.

[25] K. Palanisamy, D. Singhania, and A. Yao, "Rethinking CNN models for audio classification," *arXiv preprint arXiv:2007.11154*, 2020.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[27] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *ICML*, 2019.

[28] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," in *ICLR*, 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 *Interspeech*, 2019.

[30] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *UAI*, 2018.

[31] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[33] H. B. Sailor, D. M. Agrawal, and H. A. Patil, "Unsupervised filterbank learning using convolutional restricted boltzmann machine for environmental sound classification." in *Interspeech*, 2017.

[34] S. Majumdar and B. Ginsburg, "Matchboxnet–1d time-channel separable convolutional neural network architecture for speech commands recognition," *arXiv preprint arXiv:2004.08531*, 2020.

[35] J. Lin, K. Kilgour, D. Roblek, and M. Sharifi, "Training keyword spotters with limited and synthesized speech data," in *ICASSP*, 2020.