PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition

Preprint · December 2019

DOI: 10.48550/arXiv.1912.10211

6 authors, including:

Qiuqiang Kong
University of Surrey

Yin Cao
University of Surrey

Turab Iqbal

PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition

Qiuqiang Kong, *Student Member, IEEE*, Yin Cao, *Member, IEEE*, Turab Iqbal, Yuxuan Wang, Wenwu Wang, *Senior Member, IEEE* and Mark D. Plumbley, *Fellow, IEEE*

***Abstract*—Audio pattern recognition is an important research topic in the machine learning area, and includes several tasks such as audio tagging, acoustic scene classification and sound event detection. Recently neural networks have been applied to solve audio pattern recognition problems. However, previous systems focus on small datasets, which limits the performance of audio pattern recognition systems. Recently in computer vision and natural language processing, systems pretrained on large datasets have generalized well to several tasks. However, there is limited research on pretraining neural networks on large datasets for audio pattern recognition. In this paper, we propose large-scale pretrained audio neural networks (PANNs) trained on AudioSet. We propose to use Wavegram, a feature learned from waveform, and the mel spectrogram as input. We investigate the performance and complexity of a variety of convolutional neural networks. Our proposed AudioSet tagging system achieves a state-of-the-art mean average precision (mAP) of 0.439, outperforming the best previous system of 0.392. We transferred a PANN to six audio pattern recognition tasks and achieve state-of-the-art performance in many tasks. Source code and pretrained models have been released.**

***Index Terms*—Audio tagging, pretrained audio neural networks, transfer learning.**

I. Introduction

Our world is surrounded with sounds that contain rich information of where we are and what events are happening around us. Audio pattern recognition is an important research topic in machine learning and plays an important role in our life. Audio pattern recognition contains several tasks such as audio tagging [1], acoustic scene classification [2], sound event detection [3], classifying the emotion of patients from their speech [4], detecting the abnormal snoring of a patient [5] and classifying heart sounds.

Audio pattern recognition has attracted increasing research efforts in recent years. Early works of audio pattern recognition focused on private datasets collected by individual researchers [6], [7]. For example, Woodard [6] applied a hidden Markov model (HMM) to classify three types of sounds: wooden door open and shut, metal dropped and water poured. Recently, the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge series since 2013 [8], [9], [10], [2] have provided publicly available datasets including acoustic scene classification and sound event detection datasets for researchers to use. The DCASE challenges have attracted increasing research interests in recent years. The DCASE 2019 challenge received 311 entries in five subtasks [11].

Though there were increasing works [11] on audio pattern recognition using the DCASE challenge datasets, it is still an open question of how good an audio pattern recognition system can perform when large-scale training data is available. In computer vision, there is existing research on using the large-scale ImageNet [12] for image classification. In natural language processing, there is research on using large-scale Wikipedia data [13] to train language models. However, there is limited work [1], [14] on training systems on large-scale audio datasets, for example, the systems [1], [14] were built in 1-second level but not in clip level.

A milestone for audio pattern recognition is the release of AudioSet [1], a dataset containing over 5,000 hours of audio recordings with 527 sound classes in the released version. Instead of releasing the raw audio recordings, AudioSet released embedding features of audio clips extracted from a convolutional neural network [14]. Several works [15], [16], [17], [18], [19] have investigated building systems using the embedding features [14]. However, those methods did not work on improving systems obtaining better embedding features. In this work, we investigate AudioSet tagging with a wide range of neural network systems trained using raw audio recordings. We investigate a wide range of convolutional neural networks including [20], [21] applied on both log mel spectrograms and time-domain waveforms. Several of our proposed systems have outperformed the previous state-of-the-art systems for AudioSet tagging. In addition, an analysis of the performance and computational efficiency of audio tagging has not been done in previous works [20], [21], [14].

We call the systems trained on AudioSet *pretrained audio neural networks* (PANNs). Previous works have investigated transfer learning for audio tagging. For example, in [22], systems pretrained on the Million Song Dataset were used as feature extractors for audio clips. In [20], [23], embedding features extracted from pretrained convolutional neural networks (CNNs) were used as inputs to second-stage classifiers such as neural networks or support vector machines (SVMs). In [24], [25], systems pretrained on MagnaTagATune [26] and acoustic scene [27] datasets were finetuned on other audio tagging

Q. Kong, Y. Cao, T. Iqbal, and M. D. Plumbley are with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: q.kong@surrey.ac.uk; yin.cao@surrey.ac.uk; t.iqbal@surrey.ac.uk; m.plumbley@surrey.ac.uk).

This work was supported in part by the EPSRC Grant EP/N014111/1 “Making Sense of Sounds”, in part by the Research Scholarship from the China Scholarship Council 201406150082, and in part by a studentship (Reference: 1976218) from the EPSRC Doctoral Training Partnership under Grant EP/N509772/1. This work was supported by National Natural Science Foundation of China (Grant No. 11804365)

Y. Wang is with the ByteDance AI Lab, Mountain View, CA, USA (e-mail: wangyuxuan.11@bytedance.com).

W. Wang is with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, U.K., and also with Qingdao University of Science and Technology, Qingdao 266071, China (e-mail: w.wang@surrey.ac.uk).

tasks. However, previous works on transfer learning mainly used music datasets or were limited to smaller datasets than AudioSet. In this work, we investigate transferring PANNs to a wide range of audio pattern recognition tasks, including acoustic scene classification, general audio tagging, music classification and speech emotion classification. We investigate training a system from scratch, using a PANN as a feature extractor and fine-tuning a PANN on different tasks that have not been done in previous works. In addition, to investigate the generalization ability of PANNs, few-shot learning with PANNs using limited numbers of training samples are experimented.

This paper is organized as follows. Section II introduces audio tagging with various convolutional neural networks. Section III proposes Wavegram-CNN systems. Section IV proposes data processing techniques for AudioSet tagging. Section VI shows experimental results. Section VII concludes this work.

TABLE I
CNNs for AudioSet tagging

VGGish [1]	CNN6	CNN10	CNN14
Log-mel spectrogram 96 frames $\times$ 64 mel bins	Log-mel spectrogram 1000 frames $\times$ 64 mel bins		
3×3 @ 64 ReLU	5×5 @ 64 BN, ReLU	(3×3 @ 64 BN, ReLU) $\times 2$	(3×3 @ 64 BN, ReLU) $\times 2$
MP 2×2	Pooling 2×2		
3×3 @ 128 ReLU	5×5 @ 128 BN, ReLU	(3×3 @ 128 BN, ReLU) $\times 2$	(3×3 @ 128 BN, ReLU) $\times 2$
MP 2×2	Pooling 2×2		
(3×3 @ 256 ReLU) $\times 2$	5×5 @ 256 BN, ReLU	(3×3 @ 256 BN, ReLU) $\times 2$	(3×3 @ 256 BN, ReLU) $\times 2$
MP 2×2	Pooling 2×2		
(3×3 @ 512 ReLU) $\times 2$	5×5 @ 512 BN, ReLU	(3×3 @ 512 BN, ReLU) $\times 2$	(3×3 @ 512 BN, ReLU) $\times 2$
MP 2×2 Flatten	Global pooling		Pooling 2×2
(FC 4096 ReLU) $\times 2$	FC 512, ReLU		(3×3 @ 1024 BN, ReLU) $\times 2$
FC 527, Sigmoid	FC 527, Sigmoid		Pooling 2×2
			(3×3 @ 2048 BN, ReLU) $\times 2$
			Global pooling
			FC 2048, ReLU
			FC 527, Sigmoid

II. Audio tagging systems

Audio tagging is an essential task of audio pattern recognition with the goal of predicting the presence or absence of sound events in an audio clip. Early work in audio tagging includes using manually-design features as input, such as audio energy, zero-crossing rate, and mel-frequency cepstrum coefficients (MFCCs) [28]. Generative models including Gaussian mixture models (GMMs) [29], [30], hidden Markov models (HMMs) and SVMs [31] have been used as classifiers. Recently, neural network based methods including CNNs have been used [32] to predict the tags of a music clip. CNN based systems have achieved state-of-the-art performance in several DCASE challenge tasks including acoustic scene classification [2] and sound event detection [3]. However, many of those works focused on particular tasks with a limited number of sound classes and were not designed to recognize a wide range of sound classes. In this work, we focus on training large-scale PANNs on AudioSet [1] to solve the general audio tagging problem.

A. CNNs

1) Conventional CNNs: CNNs are classes of neural networks that have been successfully applied to computer vision tasks such as image classification [33], [34] and were later adopted for audio tagging [32], [19]. Log mel spectrograms are usually used as input to CNNs [32], [19]. To begin with, the short time Fourier transform (STFT) is applied on time-domain waveforms to calculate spectrograms. Then mel filter banks are applied on the spectrograms followed by a logarithmic operation to extract log mel spectrograms [32], [19]. A CNN consists of several convolutional layers. Each convolutional layer contains several kernels that are convolved with the input to capture local patterns of input.

2) Adapting CNNs for AudioSet tagging: CNNs in this section are based on our previously proposed cross-task CNN systems for the DCASE 2019 challenge [35]. Different from [35], an extra fully-connected layer is added to the penultimate layer of CNNs which further increase the representation ability of CNNs. Therefore we have 6-, 10- and 14-layer CNNs. The 6-layer CNN consists of 4 convolutional layers with kernel sizes of 5×5, originated from AlexNet [36]. The 10- and 14-layer CNN inspired from [37] consists of 4 and 6 convolutional blocks, respectively. Each convolutional block consists of 2 convolutional layers with kernel sizes of 3×3. Batch normalization [38] is applied between each convolutional layer and ReLU nonlinearity [39] to speed up and stabilize training. Downsampling with average pooling size of 2×2 has been shown to outperform max pooling with size of 2×2 in the DCASE 2019 challenge tasks [35]. Therefore we adopt average pooling with sizes of 2×2 for downsampling after each convolutional block.

Global pooling is applied after the last convolutional layer to summarize feature maps of audio clips with various durations to fixed-length vectors. In [21], maximum and average operation are used for global pooling. We sum the averaged and maximized vectors to calculate a fixed length vector. In [35], those fixed-length vectors were used as embedding features for audio clips. Different from [35], our embedding features are obtained by adding an extra fully-connected layer to the fixed length vectors. For a particular audio pattern recognition task, a linear classifier is applied on the embedding features followed by either a softmax nonlinearity for classification tasks or sigmoid nonlinearity for tagging tasks. Dropout [40] is applied after downsampling with the global pooling layers to prevent systems from overfitting. Table I summarizes the proposed CNN systems. The number after symbol @ indicates the number of feature maps. The first column shows the VGGish network proposed by [14]. MP is the abbreviation of max pooling. In [14], an audio clip was split into 1-second segments and assumed each segment inherits the label of the audio clip. In contrast, our systems shown from the second to the fourth column of Table I are trained on audio clips directly without requiring the segment-wise labels.

We denote the input representation of an audio clip as x_n, where n is the index of an audio clip, and $f(x_n) \in [0, 1]^K$ is the output of the neural network representing the presence

TABLE II
RESNETS FOR AUDIOSET TAGGING

ResNet22	ResNet38	ResNet54
Log mel spectrogram 1000 frames ×64 mel bins		
$(3 \times 3$ @ $512, \text{BN}, \text{ReLU}) \times 2$		
Pooling 2×2		
(BasicB @ 64) × 2	(BasicB @ 64) × 3	(BottleneckB @ 64) × 3
Pooling 2×2		
(BasicB @ 128) × 2	(BasicB @ 128) × 4	(BottleneckB @ 128) × 4
Pooling 2×2		
(BasicB @ 256) × 2	(BasicB @ 256) × 6	(BottleneckB @ 256) × 6
Pooling 2×2		
(BasicB @ 512) × 2	(BasicB @ 512) × 3	(BottleneckB @ 512) × 3
Pooling 2×2		
$(3 \times 3$ @ $2048, \text{BN}, \text{ReLU}) \times 2$		
Global pooling		
FC 2048, ReLU		
FC 527, Sigmoid		

probability of K sound classes in the audio clip. The target is denoted as $y \in \{0,1\}^K$. The loss function l for AudioSet tagging is calculated using binary cross-entropy:

$$l = -\sum_{n=1}^{N}(y_n \ln f(x_n) + (1 - y_n)\ln(1 - f(x_n)), \quad (1)$$

where N is the number of training samples. In training, the parameters of the neural network $f(\cdot)$ are optimized by minimizing the loss function l.

B. ResNets

1) Conventional ResNets: Deeper CNNs have been shown to achieve better performance than shallower CNNs [33]. However, one challenge of very deep conventional CNNs is that the gradients do not propagate properly [34]. To solve this problem, ResNets [34] were proposed to introduce shortcut connections between convolutional layers. In this way the forward and backward signals can be directly propagated from one layer to any other layers. The shortcut connections introduce neither extra parameters nor additional computational complexity. A ResNet consists of several blocks, where each block consists of two convolutional layers with kernel sizes of 3×3 and a shortcut connection. Bottleneck blocks can be used to replace the basic blocks in a ResNet, where each bottleneck block consists of three convolutional layers with a network-in-network architecture [41].

2) Adapting ResNets for AudioSet tagging: We adapt ResNet [34] for AudioSet tagging as follows. To begin with, two convolutional layers and a downsampling layer are applied on the log mel spectrogram to reduce the input size. We implement a 22-layer ResNet with 8 basic blocks. Furthermore, we implement a 38-layer and a 54-layer ResNet with basic blocks and residual blocks, respectively. Table II shows the architecture of ResNet systems for AudioSet tagging. The BasicB and BottleneckB are abbreviations of basic block and bottleneck block, respectively.

C. MobileNets

1) Conventional MobileNets: CNNs and ResNets introduced in the above subsections contain a large amount of parameters and multiply-add operations. The computational complexity is an important issue when systems are implemented to portable devices. For example, in robotics, self-driving cars and augmented reality, the recognition tasks need to be carried out in a timely fashion on a computationally-limited platform [42]. MobileNets were designed to reduce the number of parameters and multiply-add operations in a CNN. MobileNets are based on depthwise separable convolutions by factorizing a standard convolution into a depthwise convolution and a 1×1 pointwise convolution [42]. MobileNetV1 was later improved by MobileNetV2 [43] by introducing inverted residuals to the residual blocks, which further reduced the number of parameters.

TABLE III
MOBILENETS FOR AUDIOSET TAGGING

MobileNetV1	MobileNetV2
3×3 @ 32, BN, ReLU	
Pooling 2×2	
V1Block @ 64 V1Block @ 128	V2Block, t=1 @ 16 (V2Block, t=6 @ 24) × 2
Pooling 2×2	
V1Block @ 128 V1Block @ 256	(V2Block, t=6 @ 32) × 3
Pooling 2×2	
V1Block @ 256 V1Block @ 512	(V2Block, t=6 @ 64) × 4
Pooling 2×2	
(V1Block @ 512) × 5 V1Block @ 1024	(V2Block, t=6 @ 96) × 3
Pooling 2×2	
V1Block @ 1024	(V2Block, t=6 @ 160) × 3 (V2Block, t=6 @ 320) × 1
Global pooling	
FC, 1024, ReLU	
FC, 527, Sigmoid	

2) Adapting MobileNets for AudioSet tagging: We adapt MobileNetV1 [42] and MobileNetV2 [43] systems for AudioSet tagging. Table III shows the configurations of MobileNetV1 and MobileNetV2 for AudioSet tagging. V1Block and V2Blocks are MobileNet convolutional blocks [42], [43] and consists of two and three convolutional layers, respectively.

D. One-dimensional CNNs

Previous audio tagging systems are based on the log mel spectrogram trained with CNN systems. However, the log mel spectrogram is also a hand-crafted feature. There were several works proposed to build one-dimensional CNNs on the time-domain waveforms. For example, Dai et al. proposed a one-dimensional CNN [33] for solving acoustic scene classification problems. Lee et al. proposed a one-dimensional CNN [44], which was later adopted by [21] for music tagging.

1) DaiNet: DaiNet [33] applied a kernel with a length of 80 and a stride of 4 to the input waveform of audio recordings. The kernels are learnable during training. After the first layer, a maximum operation is applied to remove phase information. The maximum operation was designed to improve the robustness of system to the phase shift of input signals. Then, several one-dimensional convolutional blocks with kernel sizes of 3 and strides of 4 were applied to extract high level features. In [33] an 18-layer DaiNet with four convolutional layers in each convolutional block had achieved the best classification performance on the UrbanSound8K dataset.

2) LeeNet: Different from the DaiNet that applied large kernels in the first layer, LeeNet [44] applied small kernels with a length of 3 on the waveforms. LeeNet stacked the kernels with lengths of 3 to replace the fixed length windows for STFT. LeeNet is composed of several convolutional layers with kernel sizes of 3. Each convolutional layer was followed by a downsampling layer with a size of 2. The original LeeNet consists of 11 layers.

3) Adapting one-dimensional CNNs for AudioSet tagging: We improve the LeeNet to a deeper architecture that has 24 layers by replacing each convolutional layer with a convolutional block that consists of two convolutional layers. To further increase the number of layers of the one-dimensional CNNs, we propose *Res1dNet* based on LeeNet with small kernel sizes of 3. We replace the convolutional blocks in LeeNet with residual blocks, where each basic residual block consists of two convolutional layers with kernel sizes of 3. The first and second convolutional layers of a convolutional block have dilations of 1 and 2 respectively to increase the receptive field of a residual block. Downsampling is applied after each residual block. By using 14 and 24 residual blocks, we obtain a Res1dNet31 and a Res1dNet51 with 31 and 51 layers, respectively.

III. Proposed Wavegram-CNN systems

The previously proposed one-dimensional CNN systems [33], [44], [21] do not outperform the systems using log mel spectrograms as input. One disadvantage of the previously proposed time-domain CNN systems [33], [44] is that they are not designed to capture frequency information. For example, the one-dimensional CNNs are not invariant to frequency shift of sound events, because the filters of one-dimensional CNNs only have one dimension that captures the time dependency but not the frequency information. In this section, we propose a novel Wavegram-CNN and a Wavegram-Logmel-CNN architecture for AudioSet tagging. Frequency patterns are important for audio pattern recognition. A classifier should be robust to the frequency shift of sound events. A melody played by a piano with shifted pitch should also be recognized as a piano. On the other hand, the frequency patterns of sound events can be learned by a two-dimensional CNN using log mel spectrograms as input. The disadvantage of log mel spectrograms is that the STFT and mel filter banks are manually selected pre-processing procedures which might not be optimal for audio pattern recognition. The one-dimensional CNNs do not require hand-crafted transforms and learn from time-domain waveforms directly. This can learn representations that are not captured by human. To solve this problem, we propose a Wavegram-CNN system to combine the advantage of a one-dimensional CNN and a two-dimensional CNN.

A. Wavegram-CNN systems

First, we apply a one-dimensional CNN to a time-domain waveform. The one-dimensional CNN begins with a convolutional layer with a filter length of 11 and a stride of 5 to reduce the size of the input. This immediately reduces the input lengths by 5 times to reduce memory usage. Then, three convolutional blocks are applied where each convolutional block consists of two convolutional layers with dilations of 1 and 2, respectively. Each convolutional block is followed by a downsampling layer with a ratio of 4. By choosing these downsampling ratios, there are 100 frames of features in one second. We denote the output size of the one-dimensional CNN layers as $T \times C$, where T is the number of frames and C is the number of channels. We reshape this output to a tensor with a size of $T \times F \times C/F$, which is called *Wavegram* where F is the number of frequency bins in the Wavegram. This design ensures the Wavegram to learn frequency patterns and be invariant to frequency shift introduced by the application of a two-dimensional CNN on the Wavegram. The two-dimensional CNN can be any architectures in Section II. In the following experiments, we adopt CNN14 as the two-dimensional CNN for Wavegram-CNN for its simplicity and robustness in AudioSet tagging.

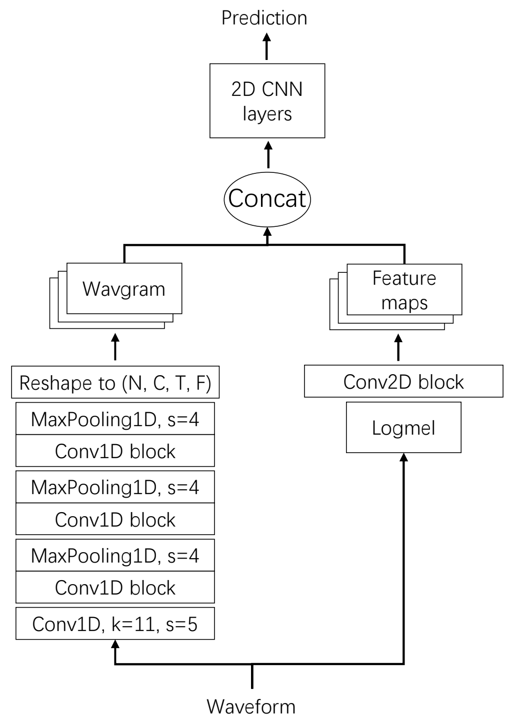

Fig. 1. Architecture of Wavegram-Logmel-CNN

B. Wavegram-Logmel-CNN

In addition to the Wavegram-CNN, we combine the Wavegram and the log mel spectrogram as inputs to a two-dimensional CNN. This can utilize the information from both time-domain waveforms and log mel spectrograms. The Wavegram can be concatenated with feature maps of a log mel spectrogram extracted by convolutional layers. The Wavegram provides extra information for audio tagging as a complement to the log mel spectrogram. Figure 1 shows the architecture of the Wavegram-Logmel-CNN.

IV. Data processing

In this section, we introduce data processing for AudioSet tagging, including data balancing and data augmentation. Data balancing is a technique to solve the training problem of a highly unbalanced dataset. Data augmentation is another technique to enlarge the dataset to prevent systems from overfitting during training.

A. Data balancing

The number of audio clips available for training for different sound classes is different in AudioSet. For example, there are over 900,000 audio clips belonging to the categories "Speech" and "Music". On the other hand, there are only tens of audio clips belonging to the category "Toothbrush", for example. The number of audio clips for different sound classes has a long tail distribution. In training an AudioSet tagging system, the data is input to the neural network in mini-batches. Without the balancing strategy, audio clips are uniformly sampled from AudioSet for training. The sound classes with more samples such as "Speech" are more likely to be sampled for training. In an extreme case, all samples in a mini-batch belong to the same sound class. This will cause the system to overfit to sound classes with more samples and underfit to sound classes with fewer samples. To solve this problem, we designed a balanced sampling strategy containing two stages in training. The first stage uniformly selects sound classes for training. The second stage samples audio clips from the selected sound classes. In each mini-batch, the numbers of audio clips for different sound classes are almost the same. We use the term "almost" because there can be multiple tags for an audio clip. Therefore, each mini-batch is almost balanced instead of strictly balanced.

B. Data augmentation

Data augmentation is a useful way to prevent a system from overfitting. Some sound classes in AudioSet have few training samples which may limit the audio tagging performance. We applied Mixup [45] and SpecAugment [46] for AudioSet augmentation.

1) Mixup: Mixup [45] is a way to augment a dataset by interpolating both the input and target of two samples from a dataset. For example, we denote the input as x_1, x_2 and the target as y_1, y_2 of two samples, respectively. Then, the augmented sample can be obtained by $x = \lambda x_1 + (1 - \lambda) x_2$ and $y = \lambda y_1 + (1 - \lambda) y_2$, where λ is sampled from a Beta distribution. Mixup was shown to reduce the memorization of corrupt labels and increase the robustness to adversarial examples [45]. In AudioSet tagging, we only apply mixup to the samples that have been used for training at least once. We do not apply mixup to the samples that have not been used for training.

2) SpecAugment: SpecAugment [46] was proposed for augmenting speech data for speech recognition. SpecAugment is operated on the log mel spectrogram of an audio clip and contains frequency masking and time masking. The frequency masking is applied such that f consecutive mel frequency channels $[f_0, f_0 + f]$ are masked, where f is chosen from a uniform distribution from 0 to the frequency mask parameter f', and f_0 is chosen from $[0, F - f]$, where F is the number of mel frequency channels [46]. There can be more than one frequency mask in each log mel spectrogram. The frequency mask can improve the robustness of the systems to frequency distortion of audio clips recorded with different devices. Time masking is similar to frequency masking but is applied in the time domain.

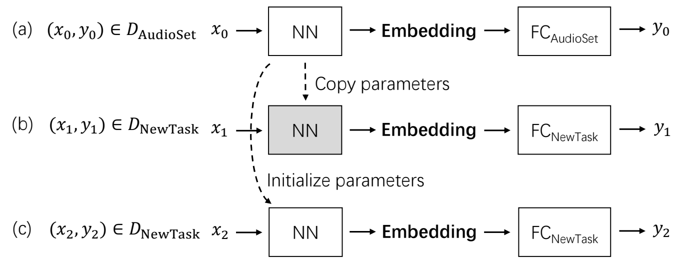

Fig. 2. (a) A PANN is pretrained with the AudioSet dataset. (b) For a new task, the PANN is used as the feature extractor. A classifier is built on the extracted embedding features. The shadow rectangle indicates the parameters are frozen and not trained. (c) For a new task, the parameters of a neural network are initialized with a PANN. Then, fine-tune all parameters of the neural network on the new task.

V. Transfer to other tasks

We call the audio tagging systems pretrained on AudioSet as PANNs. To investigate the generalization ability of PANNs, we propose to transfer PANNs to a wide range of audio pattern recognition tasks. Previous works [22], [20], [23], [24], [26] of transfer learning mainly focused on music tagging, and were limited to smaller datasets than AudioSet. In addition, there is a lack of research on comparing the performance of different transfer learning methods including using pretrained neural networks as feature extractors and fine-tuning pretrained neural networks. In this work, we propose the following transfer learning strategies.

1) Training a system from scratch. The architecture of a PANN is used for a new task except the final fully-connected layer. All parameters are randomly initialized. No information from the AudioSet dataset is used for training the new task. This training from scratch system is used as a baseline system to be compared with other transfer learning strategies.

2) Using a PANN as a feature extractor. A PANN is used as a feature extractor for a new task. First, the embedding features of an audio clip are obtained from the penultimate layer of the PANN. Then, the embedding features are used as input to another classifier such as a fully-connected neural network. In training, the parameters of the PANN are not trained. Only the parameters of the classifier built on the embedding features are trained. Fig. 2(a) shows training a PANN with the AudioSet dataset. Fig. 2(b) shows transfer learning with using the PANN as a feature extractor. The D_{AudioSet} and D_{NewTask} denotes the AudioSet dataset and a new task dataset. The $\text{FC}_{\text{AudioSet}}$ and $\text{FC}_{\text{NewTask}}$ denote the fully connected layer for AudioSet tagging and the classification or tagging of a new task. The input and prediction are denoted as x_j and y_j, respectively, where j is the index of task.

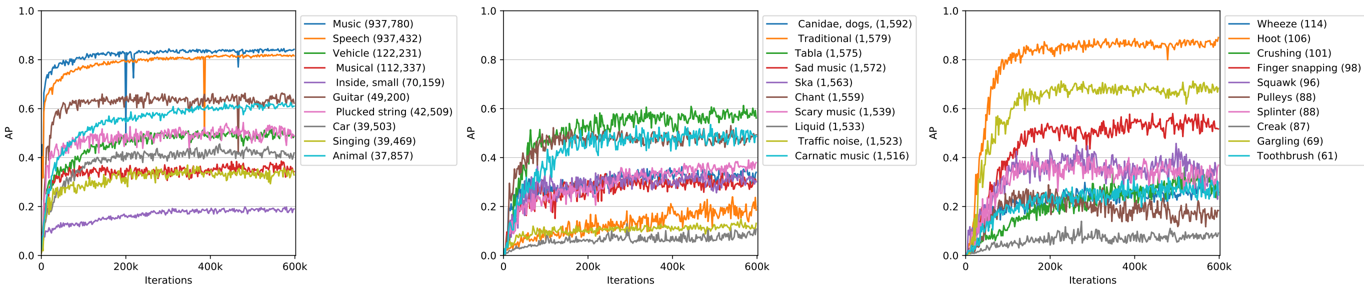

Fig. 3. Class-wise AP of sound events with CNN14 system. The number inside parentheses indicates the number of training samples. The left, middle and right columns show the AP of sound classes sorted by number of training samples in descending order. The left, middle, right columns show the number of training samples ranked the 1-10, 250-260 and 517-527 in the training set of AudioSet.

(3) Fine-tuning a PANN. The architecture of a PANN is used for a new task except the final fully-connected layer. All parameters to calculate the embedding features are initialized from the PANN instead of being randomly initialized. The fully connected layers after the embedding features are randomly initialized. During the training on a new task, all parameters are fine-tuned. Fig. 2(c) shows the diagram of fine-tuning a PANN.

VI. Experiments

First, we evaluate AudioSet tagging performance with the proposed systems. Then, the PANNs are transferred to several audio pattern recognition datasets including acoustic scene classification, general audio tagging, music classification and speech emotion classification.

A. AudioSet

AudioSet is a large-scale audio dataset with an ontology of 527 sound classes [1]. The audio clips from AudioSet are extracted from YouTube videos. The training set consists of 2,063,839 audio clips including a balanced subset of 22,160 audio clips. The evaluation set consists of 20,371 audio clips. Instead of using the embedding features provided by [1], we download the audio clips from the internet using the links provided by [1] and skip the audio clips that are not downloadable. We successfully downloaded 1,934,187 audio clips in the full training set including 20,550 audio clips in the balanced training set. We successfully downloaded 18,887 audio clips in the evaluation dataset. We padded the audio clips to 10 seconds with silence if they were shorter than 10 seconds. Considering the fact that a large amount of audio clips from YouTube are monophonic and have a low sampling rate, we convert all audio clips to monophonic and resampled them to 32 kHz.

We extract log mel spectrograms for the CNN systems. STFT is applied on the waveform with a hamming window size of 1024 [35] and a hop size of 320 samples to obtain the spectrograms. This leads to 100 frames in a second mel spectrogram. We compare using different hop sizes in our experiments. We apply 64 mel filter banks to calculate the log mel spectrogram following [35]. The cut off frequencies of the mel banks are 50 Hz and 14 kHz to remove low frequency noise and aliasing effects. The log mel spectrogram of a 10-second audio clip has a shape of 1000×64.

TABLE IV
Comparison with previous methods

	mAP	AUC	d-prime
Random guess	0.005	0.500	0.000
Google CNN [1]	0.314	0.959	2.452
Single-level attention [15]	0.337	0.968	2.612
Multi-level attention [16]	0.360	0.970	2.660
Large feature-level attention [19]	0.369	0.969	2.640
TAL Net [18]	0.362	0.965	2.554
DeepRes [47]	0.392	0.971	2.682
Our proposed CNN14	**0.431**	**0.973**	**2.732**

B. Evaluation metrics

Mean average precision (mAP), mean area under the curve (mAUC) and d-prime are used as official evaluation metrics for AudioSet tagging [1]. Average precision (AP) is the area under the recall and precision curve. AP does not depend on the number of true negatives because neither precision nor recall depends on the number of true negatives. On the other hand, AUC is the area under the false positive rate and true positive rate (recall) which reflects the influence of the true negatives. The d-prime can be calculated from AUC [1]. All metrics are calculated on individual classes and then averaged across all classes. These metrics are also called macro metrics.

C. AudioSet tagging results

1) Comparison with previous methods: Table IV shows the comparison of previous AudioSet tagging methods and our proposed CNN14. The random guess achieves an mAP of 0.005, an AUC of 0.500 and a d-prime of 0.000. The result released by Google [1] trained with embedding features from [14] achieved an mAP of 0.314 and an AUC of 0.959. The single-level attention and multi-level attention systems achieved mAPs of 0.337 and 0.360, which were later improved by a large feature-level attention neural network that achieved an mAP of 0.369. Wang et al. [18] investigated five different types of attention functions and achieved an mAP of 0.362. All the above systems were built on the embedding features released by [1]. The recent DeepRes system [47] built audio tagging systems on waveforms downloaded from YouTube and have achieved an mAP of 0.392. The bottom row shows our proposed CNN14 system achieves an mAP of 0.431 and

TABLE V
RESULTS OF DIFFERENT HOP SIZES

Hop size	Time resolution	mAP	AUC	d-prime
1000	31.25 ms	0.400	0.969	2.645
640	20.00 ms	0.417	0.972	2.711
500	15.63 ms	0.417	0.971	2.682
320	10.00 ms	**0.431**	**0.973**	**2.732**

TABLE VI
RESULTS OF DIFFERENT EMBEDDING DIMENSIONS

Embedding dimension	mAP	AUC	d-prime
32	0.364	0.958	2.437
128	0.412	0.969	2.634
512	0.420	0.971	2.689
2048	**0.431**	**0.973**	**2.732**

TABLE VII
RESULTS OF AUDIOSET TAGGING SYSTEMS

	mAP	AUC	d-prime
CNN6	0.343	0.965	2.568
CNN10	0.380	0.971	2.678
CNN14	**0.431**	**0.973**	**2.732**
ResNet22	0.430	0.973	0.270
ResNet38	**0.434**	**0.974**	**2.737**
ResNet54	0.429	0.971	2.675
MobileNetV1	**0.389**	**0.970**	**2.653**
MobileNetV2	0.383	0.968	2.624
DaiNet [33]	0.295	0.958	2.437
LeeNet11 [44]	0.266	0.953	2.371
LeeNet18	0.336	0.963	2.525
Res1dNet31	**0.365**	**0.958**	**2.444**
Res1dNet51	0.355	0.948	2.295
Wavegram-CNN	0.389	0.968	2.612
Wavegram-Logmel-CNN	**0.439**	**0.973**	**2.720**

outperforms the previous state-of-the-art systems by a large margin.

2) Class-wise performance: Fig. 3 shows the class-wise AP of different sound classes with the CNN14 system. The left, middle and right columns shows the AP of sound classes sorted by the number of training samples in descending order. The performance of different sound classes can be very different. For example, “Music” and “Speech” achieve APs of over 0.8. On the other hand, some sound classes such as “Inside, small” achieve an AP of only 0.19. Fig. 3 shows that the APs are usually not correlated with the number of training samples. For example, the left column shows that “Inside, small” contains 70,159 training samples while its AP is low. On the other hand, the right column shows that “Hoot” only has 106 training samples but achieves an AP of 0.86 that is larger than many other sound classes with more training samples. Fig. 11 shows the class-wise comparison of CNN14, MobileNetV1 system with the previous audio tagging system [19] built with embedding features extracted by [1]. The blue bars show the number of training samples in logarithmic scale. The plus symbols indicate label quality between 0 and 1. The red symbols indicate the classes where label quality is not available. Fig. 11 shows that our proposed CNN14 outperform the system in [19] in a wide range of sound classes.

3) Hop sizes: Hop size is the number of samples between adjacent frames. A smaller hop size leads to higher resolution in the time domain. Hop size is a hyper-parameter which controls the time domain resolution of log mel spectrograms. We investigate the influences of different hop sizes on AudioSet tagging with the CNN14 system. Hop sizes of 1000, 640, 500 and 320 are experimented which correspondingly result in time domain resolutions of 31.25 ms, 20.00 ms, 15.63 ms and 10.00 ms between adjacent frames. Table V shows that the mAP increases with smaller hop sizes. The system with a hop size of 320 achieves an mAP of 0.431, outperforming the other hop sizes of 500, 640 and 1000.

4) Embedding dimensions: Embedding features are fixed-length vectors that summarize an audio clip. The embedding feature vector of the previous CNN14 system has a dimension of 2048. In some applications, compact embeddings with smaller dimensions are required for computational efficiency. We investigate building CNN14 with embedding dimensions of 32, 128, 512 and 2048, respectively. Table VI shows that with an embedding dimension of 2048, an mAP of 0.431 is achieved. The mAPs slightly decrease to 0.420 and 0.412 with embedding dimensions of 512 and 128. The mAP drops to 0.364 with an embedding dimension of 32.

5) Number of CNN layers: We investigate the performance of CNNs with 6, 10 and 14 layers described in Section II-A. Table VII shows that the 6-, 10- and 14-layer CNNs achieve mAPs of 0.343, 0.380 and 0.431, respectively. In contrast to the results that deeper CNNs with more than 9 layers underperform CNN with 9 layers trained on the DCASE datasets [35], we show that with the large-scale AudioSet dataset the 14-layer CNN system can outperform the 10-layer in mAP by a large margin. Results show that with the increased number of convolutional layers, the performance of systems improves. This is because large-scale dataset is able to make advantage of deeper CNNs without overfitting, whereas smaller datasets suffer from overfitting, so shallower CNNs achieve better performance for small-scale dataset.

6) ResNets: To further investigate the AudioSet tagging performance with the increased number of convolutional layers, we evaluate the AudioSet tagging performance with ResNets. Table VII shows that ResNet22 achieves an mAP of 0.430 that is similar to the CNN14 system. ResNet38 achieves an mAP of 0.434 which slightly outperforms other systems. ResNet54 achieves an mAP of 0.429, which does not further improve the performance and may be caused by overfitting.

7) MobileNets: Previous systems show that the CNN systems have achieved good performance in AudioSet tagging. However, those systems do not consider the computational efficiency when systems are implemented in portable devices. We investigate the performance of AudioSet tagging with light weight MobileNets. Table VII shows that MobileNetV1 achieves an mAP of 0.389, which is 0.042 lower than the CNN14 system. The number of multiplication and addition operations (multi-adds) and parameters of the MobileNetV1 are only 8.6% and 5.9% of the CNN14 system. The MobileNetV2 achieves an mAP 0f 0.383 and is more computationally

TABLE VIII
NUMBER OF MULTI-ADDS AND PARAMETERS OF DIFFERENT SYSTEMS

	Multi-Adds	Parameters
CNN6	21.986 G	4,837,455
CNN10	21.986 G	4,837,455
CNN14	42.220 G	80,753,615
ResNet22	30.081 G	63,675,087
ResNet38	48.962 G	73,783,247
ResNet54	54.563 G	104,318,159
MobileNetV1	3.614 G	4,796,303
MobileNetV2	2.810 G	4,075,343
DaiNet	30.395 G	4,385,807
LeeNet	4.741 G	748,367
LeeNet24	26.369 G	10,003,791
Res1dNet31	32.688 G	80,464,463
Res1dNet51	61.833 G	106,538,063
Wavegram-CNN	44.234 G	80,991,759
Wavegram-Logmel-CNN	53.510 G	81,065,487

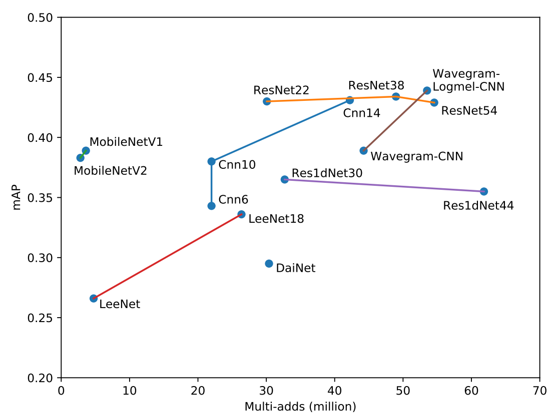

Fig. 4. Multi-adds versus mAP of AudioSet tagging systems. The same types of architectures are grouped in the same color.

efficient than MobileNetV1, where the number of multi-adds and parameters are only 6.7% and 5.0% of CNN14 which greatly reduces the computational cost.

8) One-dimensional CNNs: Table VII shows the performance of one-dimensional CNNs. The DaiNet with 18 layers [33] achieves an mAP of 0.295. The LeeNet11 with 11 layers [44] achieves an mAP of 0.266. Our extended LeeNet with 24 layers improves the mAP from 0.266 to 0.336. Furthermore, our proposed Res1dNet31 and Res1dNet51 systems achieve mAPs of 0.365 and 355, respectively, and have achieved the state-of-the-art performance among the one-dimensional CNN systems.

9) Wavegram-CNN: The bottom rows of Table VII show the result of our proposed Wavegram-CNN and Wavegram-Logmel-CNN systems. The Wavegram-CNN system achieves an mAP of 0.389, outperforming the best previous one-dimensional CNN system ResNet30 of 0.365. This indicates that the proposed combination of one-dimensional and two-dimensional CNNs are effective in audio pattern recognition. Furthermore, our proposed Wavegram-Logmel-CNN system achieves a state-of-the-art result among all AudioSet tagging systems with mAP, mAUC and d-prime of 0.439, 0.973 and 2.720, respectively.

10) Complexity analysis: We analyze the computational complexity of the aforementioned pattern recognition systems. The number of multi-adds and parameters are two important factors for systems running on portable devices. The middle column of Table VIII shows the number of multi-adds to infer a 10-second audio clip. The unit prefix "G" indicates 10^9. The right column of Table VIII shows the number of parameters of different systems. The number of multi-adds and parameters of the CNN14 system are 42.220 G and around 81 million, which are larger than the CNN6 and CNN10 systems. The number of multi-adds of the ResNets22 and ResNet38 are slightly less than the CNN14 system. The ResNet54 system contains the most multi-adds of 54.563 G. The one-dimensional CNNs have a similar computational cost as the two-dimensional CNNs. The best one-dimensional system Res1dNet31 contains 32.688 G multi-adds and around 80 million parameters. Our proposed Wavegram-CNN contains 44.234 G multi-adds and around 81 million parameters. The Wavegram-Logmel-CNN slightly increases the multi-adds to 53.510 G. The number of parameters is around 81 million, which is not much more. To reduce the number of multi-adds and parameters, MobileNets are applied. The MobileNetV1 and MobileNetV2 are light weight CNNs, with only 3.614 G and 2.810 G multi-adds and around 5 million and 4 million parameters, respectively. The MobileNets greatly reduce the computational cost and the system size while minimizing the impact on performance. Figure 4 shows the multi-adds versus mAP of different AudioSet tagging systems. The same type of systems are grouped in the same color. The multi-adds increases from left to right. The mAP increases from bottom to top. On the top-right is our proposed Wavegram-Logmel-CNN system that achieves the best mAP. On the top-left are MobileNetV1 and MobileNetV2 that are the most computationally efficient systems that do not reduce the mAP much.

D. Transfer to other tasks

In this section we apply PANNs trained on AudioSet to other pattern recognition tasks. We resampled audio recordings to 32 kHz and converted audio recordings to monophonic to be consistent with the PANNs trained on AudioSet. We perform experiments the transfer learning strategies in Section V. For the strategy using a PANN as the feature extractor, we call the system using one and three fully-connected layers on the embedding features Freeze_L1 and Freeze_L3, respectively. We adopt CNN14 as the base system for transfer learning to other tasks for its simplicity and robustness to AudioSet tagging. Few-shot learning is an important issue in audio pattern recognition. This is because for some tasks, only a limited number of training samples are provided. To investigate the performance of the systems with few-shot learning, we evaluate the systems trained with a various number of training samples. This can show how PANNs can help audio pattern recognition using few training samples.

1) ESC-50: ESC-50 is an environmental sounds dataset [49] consisting of 50 sound events in total such as "Dog" and "Rain". There are 2,000 5-second audio clips in the

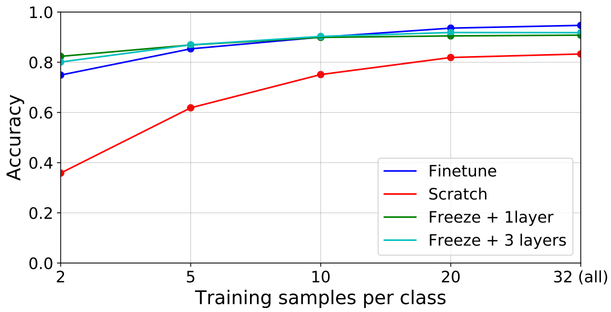

Fig. 5. Accuracy of ESC-50 with various number of training samples per class.

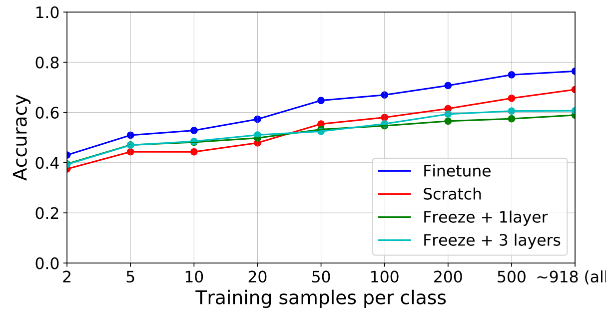

Fig. 6. Accuracy of DCASE 2019 Task 1 with various number of training samples per class.

TABLE IX
ACCURACY OF ESC-50

	STOA [48]	Scratch	fine-tune	Freeze_L1	Freeze_L3
Acc.	0.865	0.833	**0.947**	0.908	0.918

TABLE X
ACCURACY OF DCASE 2019 TASK 1

	STOA [50]	Scratch	Fine-tune	Freeze_L1	Freeze_L3
Acc.	**0.851**	0.691	0.764	0.589	0.607

dataset with 40 samples per class. Table IX shows the 5-fold cross validation [49] accuracy of systems. The Previous state-of-the-art system [48] achieved an accuracy of 0.865 using unsupervised filterbank learning which uses a convolutional restricted Boltzmann machine. Our system fine-tuned on PANN achieves an accuracy of 0.947 and outperforms the previous state-of-the-art system with a large margin. The Freeze_L1 and Freeze_L3 systems achieve an accuracy of 0.918 and 0.908, respectively. Training the system from scratch achieves an accuracy of 0.833. Fig. 5 shows the accuracy of PANNs with various numbers of training samples of each sound class. Using PANN as a feature extractor achieves the best performance when only a few samples are available such as fewer than 10 samples per sound class. When more training samples are available, the fine-tuned PANN achieves better performance. Both the fine-tuned system and the system using the PANN as a feature extractor outperform the systems trained from scratch.

2) DCASE 2019 Task 1: The DCASE 2019 Task 1 is an acoustic scene classification task [2]. The DCASE 2019 Task 1 dataset consists of over 40 hours of recordings collected from various acoustic scenes in 12 large European cities. We focus on subtask A where each audio recording has two channels and the sampling rate is 48 kHz. In the development set, there are 9185 and 4185 audio clips for training and validation, respectively. We convert stereo recordings to monophonic by averaging the stereo channels. The state-of-the-art system has achieved an accuracy of 0.851 using the combination of various classifiers that use stereo recordings as input [50]. We only use the monophonic information in our systems. CNN14 trained from scratch achieves an accuracy of 0.691. The fine-tuned PANN achieved an accuracy of 0.764. Freeze_L1 and Freeze_L2 achieve an accuracy of 0.689 and 0.607, which means they do not outperform the system trained from scratch. One explanation is that the task of acoustic scene classification is different from AudioSet tagging so using PANN as a feature extractor did not outperform training from scratch. Still, the fine-tuned system achieves the best performance. Fig. 6 shows the classification accuracy of systems with various numbers of training samples per class.

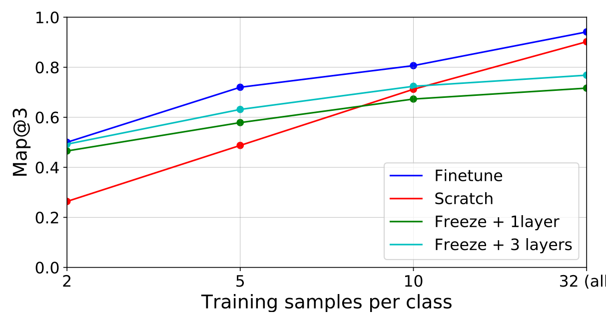

Fig. 7. Accuracy of DCASE 2018 Task 2 with various number of training samples per class.

TABLE XI
ACCURACY OF DCASE 2018 TASK 2

	STOA [51]	Scratch	Fine-tune	Freeze_L1	Freeze_L3
Acc.	**0.954**	0.902	0.941	0.717	0.768

3) DCASE 2018 Task 2: The DCASE 2018 Task 2 is a general-purpose automatic audio tagging task [53] consisting of audio recordings from Freesound annotated using a vocabulary of 41 labels from the AudioSet ontology. The development set consists of 9,473 audio recordings with durations from 300 ms to 30 s. Table XI shows that the best previous method achieved an mAP@3 of 0.954 using an ensemble of several systems [51]. The system trained from scratch achieved an accuracy of 0.902. The fine-tuned PANN achieves an mAP@3 of 0.941. The Freeze_L1 and Freeze_L3 systems achieve an accuracy of 0.717 and 0.768. Fig. 7 shows the mAP@3 with various numbers of training samples. The fine-tuned PANN outperforms the systems trained from scratch and the systems using PANN as a feature extractor. The fine-tuned PANN achieves comparable results to the state-of-the-art system.

4) MSoS: The Making Sense of Sounds (MSoS) data challenge [54] is a task to assign an audio recording to one of

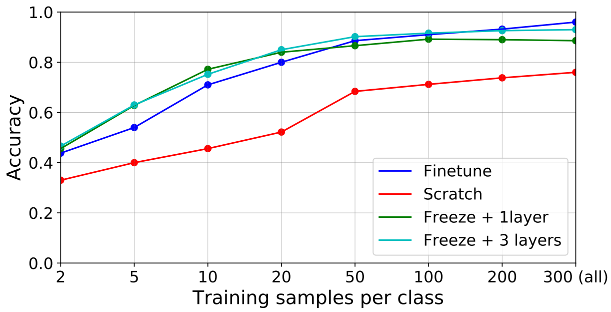

Fig. 8. Accuracy of MSoS with various number of training samples per class.

TABLE XII
ACCURACY OF MSOS

	STOA [52]	Scratch	Fine-tune	Freeze_L1	Freeze_L3
Acc.	0.930	0.760	**0.960**	0.886	0.930

five categories: "Nature", "Music", "Human", "Effects" and "Urban". The dataset consists of a development set of 1,500 audio clips and an evaluation set of 500 audio clips. All audio clips have a duration of 4 seconds. The state-of-the-art system applied transfer learning [52] and achieved an accuracy of 0.930. Our fine-tuned PANN achieves an accuracy of 0.960, which outperforms the previous state-of-the-art system. The system trained from scratch achieves an accuracy of 0.760. Fig. 8 shows the accuracy of the systems with respect to the number of training samples. With fewer samples, the fine-tuned system and the system using PANN as feature extractor outperform the system trained from scratch.

5) GTZAN: The GTZAN dataset [56] is a music genre classification dataset containing 1,000 30-second music clips of 10 genres of music such as "Classical" and "Country". All music clips have a duration of 30 seconds and a sampling rate of 22,050 Hz. In development, 10-fold cross validation is used to evaluate the performance of the systems. Table XIII shows that the previous state-of-the-art system achieved an accuracy of 0.939 using a bottom-up broadcast neural network [55]. The system fine-tuned on PANN achieves an accuracy of 0.915, outperforming the system trained from scratch with an accuracy of 0.758 and the Freeze_L1 and Freeze_L3 systems with an accuracy of 0.827 and 0.858, respectively. Fig. 9 shows the accuracy of systems with various numbers of training samples. The Freeze_L1 and Freeze_L3 systems achieve better performance than other systems when less than 10 samples per class are used as training data. With more training samples, the fine-tuned PANN performs better than other systems.

6) RAVDESS: The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) is a human speech emotion dataset [58]. The database consists of speech of 24 professional actors including 12 female and 12 male in 8 emotions such as "Happy" and "Sad". The task is to classify each speech into an emotion. There are 1,440 audio clips in the development set. We evaluate our systems with 4-fold cross validation. Table XIV shows that the previous state-of-the-art system achieved an accuracy of 0.645 using a CNN [57]. The system trained from scratch achieved an accuracy

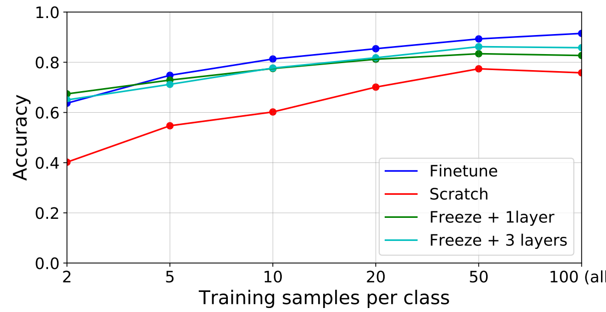

Fig. 9. Accuracy of GTZAN with various number of training samples per class.

TABLE XIII
ACCURACY OF GTZAN

	STOA [55]	Scratch	Fine-tune	Freeze_L1	Freeze_L3
Acc.	**0.939**	0.758	0.915	0.827	0.858

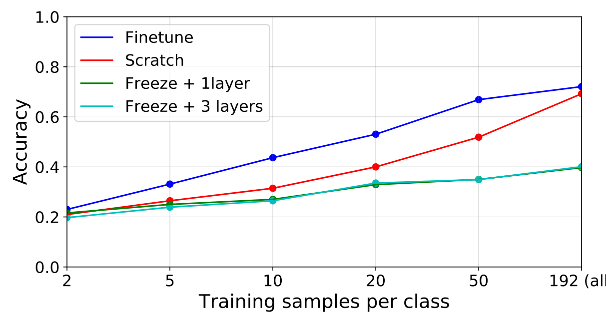

Fig. 10. Accuracy of RAVDESS with various number of training samples per class.

TABLE XIV
ACCURACY OF RAVDESS

	STOA [57]	Scratch	Fine-tune	Freeze_L1	Freeze_L3
Acc.	0.645	0.692	**0.721**	0.397	0.401

of 0.692. The system fine-tuned on PANN achieves a state-of-the-art accuracy of 0.721. The Freeze_L1 and Freeze_L3 systems achieve an accuracy of 0.397 and 0.401. Fig. 10 shows the accuracy of systems with respect to a range of training samples. The fine-tuned systems and the system trained from scratch outperform systems using PANN as a feature extractor. This indicates that systems should be fine-tuned or retrained for fine-grained classification tasks such as recognizing different emotions of speech.

E. Discussion

In this section we first evaluate the AudioSet tagging performance with different systems. Our adapted CNNs, ResNets and MobileNets have outperformed the previous state-of-the-art AudioSet tagging systems, for example, CNN14 achieves an mAP of 0.431 and ResNet38 achieves an mAP of 0.434. MobileNets are light-weight systems that have fewer multi-adds and numbers of parameters, for example, MobileNetV1 retains an mAP of 0.389 that do not reduce the performance much. Our adapted one-dimensional system Res1dNet31 achieves an mAP of 0.365, outperforming the DaiNet [33] of 0.295 and LeeNet11 [44] of 0.266. Our proposed Wavegram-CNN achieves the highest mAP of 0.389 among one-dimensional systems. Furthermore, the Wavegram-Logmel-CNN system achieves a state-of-the-art mAP of 0.439 among all systems.

The PANNs pretrained on AudioSet dataset are then transferred to six audio pattern recognition tasks. Many of previous state-of-the art systems applied the ensemble of different systems to obtain the best performance on one task. On the other hand, we do not apply ensemble of systems and only use the output of one PANN as prediction. The same PANN is used for all tasks. Still, the fine-tuned PANNs have achieved state-of-the-art performance in the ESC-50, MSOS and RAVDESS classification tasks, and have approached the state-of-the-art performance in the DCASE 2018 Task 2 and GTZAN classification tasks. All system performance increases with the number of training samples. Compared with training from scratch and using PANN as the feature extractor, the fine-tuned PANN achieves the best performance in all tasks. The experiments show that PANNs have been successful for general audio pattern recognition tasks.

The source code and pretrained models of PANNs are publicly available[1]. The source code of a demo with graphical user interface is available too[2].

VII. Conclusion

This work proposes pretrained audio neural networks (PANNs) trained on the large-scale AudioSet for audio pattern recognition. A wide range of convolutional neural networks (CNNs) are explored to solve the AudioSet tagging problem. We propose a Wavegram-Logmel-CNN that has achieved the state-of-the-art performance in AudioSet tagging. Both the AudioSet tagging performance and computational efficiency of PANNs are investigated. The PANNs are transferred to a wide range of audio pattern recognition tasks and have outperformed several previous state-of-the-art systems. The PANNs have been shown to help few-shot audio pattern recognition tasks where only limited training samples are available. In the future, we will extend the PANNs to more audio pattern recognition tasks.

References

[1] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[2] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018, pp. 9–13.

[3] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *International Joint Conference on Neural Networks (IJCNN)*, 2015.

[4] K. Dupuis and M. K. Pichora-Fuller, "Recognition of emotional speech for younger and older talkers: Behavioural findings from the toronto emotional speech set," *Canadian Acoustics*, vol. 39, no. 3, pp. 182–183, 2011.

[1]https://github.com/qiuqiangkong/audioset_tagging_cnn

[2]https://github.com/yinkalario/General-Purpose-Sound-Recognition-Demo

[5] S. J. Lim, S. J. Jang, J. Y. Lim, and J. H. Ko, "Classification of snoring sound based on a recurrent neural network," *Expert Systems with Applications*, vol. 123, pp. 237–245, 2019.

[6] J. P. Woodard, "Modeling and classification of natural sounds by product code hidden Markov models," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1833–1835, 1992.

[7] D. P. W. Ellis, "Detecting alarm sounds," in *https://academiccommons.columbia.edu/doi/10.7916/D8F19821/download*, 2001.

[8] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[9] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 2, pp. 379–393, 2018.

[10] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017, pp. 85–92.

[11] "DCASE Challenge 2019," http://dcase.community/challenge2019.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018, pp. 4171–4186.

[14] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "CNN architectures for large-scale audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 131–135.

[15] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Audio Set classification with attention model: A probabilistic perspective," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 316–320.

[16] C. Yu, K. S. Barsim, Q. Kong, and B. Yang, "Multi-level attention model for weakly supervised audio classification," in *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018, pp. 188–192.

[17] S.-Y. Chou, J.-S. R. Jang, and Y.-H. Yang, "Learning to recognize transient sound events using attentional supervision," in *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2018, pp. 3336–3342.

[18] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 31–35.

[19] Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang, and M. D. Plumbley, "Weakly labelled audioset tagging with attention neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1791–1802, 2019.

[20] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *International Society of Music Information Retrieval (ISMIR)*, 2017, pp. 141–149.

[21] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *International Society for Music Information Retrieval (ISMIR)*, 2017, pp. 637–644.

[22] A. Van Den Oord, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Conference of the International Society for Music Information Retrieval (ISMIR)*, 2014, pp. 29–34.

[23] Y. Wang, "Polyphonic sound event detection with weak labeling," *PhD thesis, Carnegie Mellon University*, 2018.

[24] J. Pons and X. Serra, "MUSICNN: Pre-trained convolutional neural networks for music audio tagging," *arXiv preprint arXiv:1909.06654*, 2019.

[25] A. Diment and T. Virtanen, "Transfer learning of weakly labelled audio," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 6–10.

[26] E. Law and L. Von Ahn, "Input-agreement: a new mechanism for collecting data using human computation games," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009, pp. 1197–1206.

[27] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *2016 24th European*

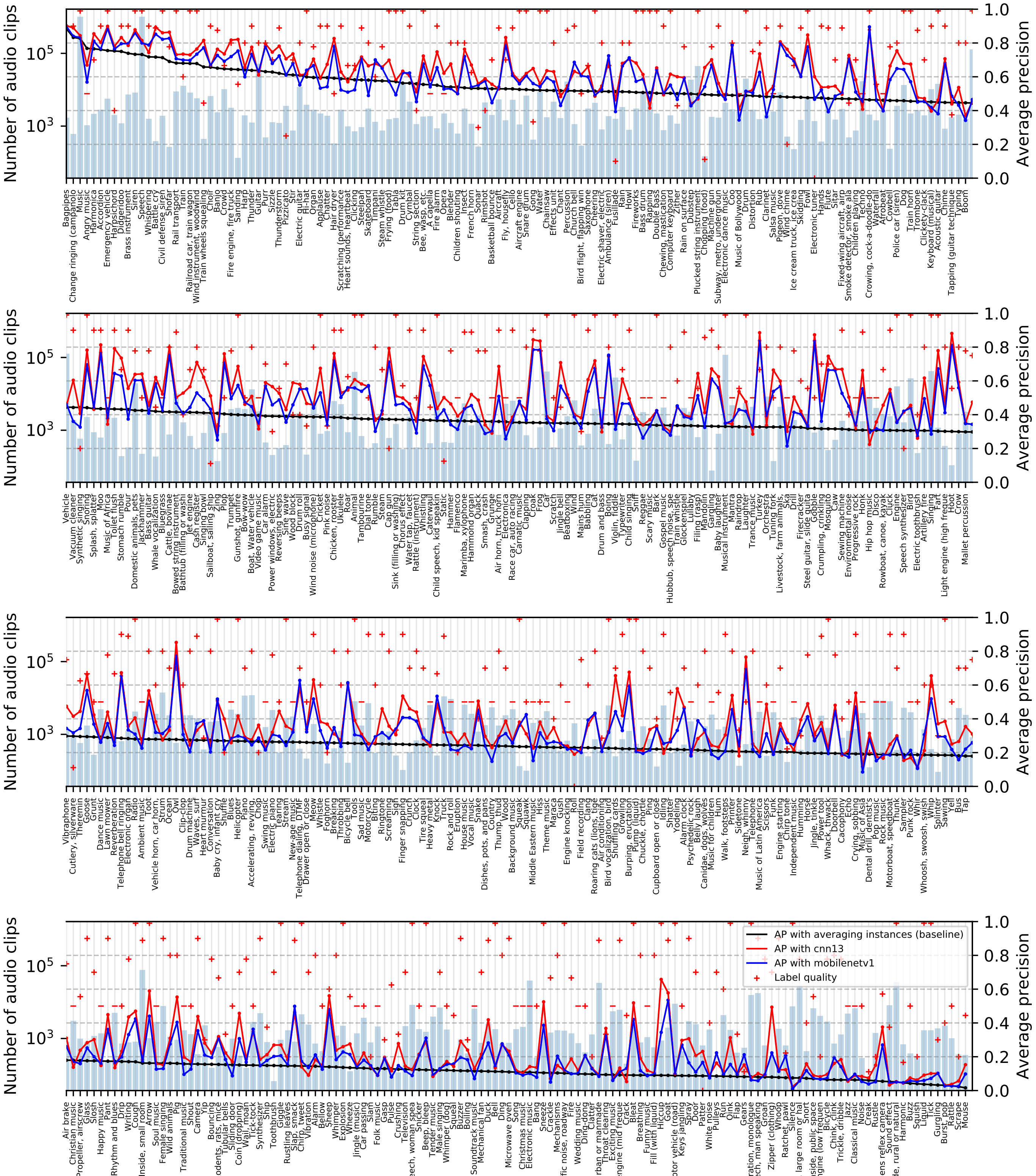

Fig. 11. Class-wise performance of AudioSet tagging systems. Red, blue and black curves are APs of CNN14, MobileNetV1 and the audio tagging system [19] built with embedding features extracted by [1].

Signal Processing Conference (EUSIPCO). IEEE, 2016, pp. 1128–1132.

[28] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee, "Classification of general audio data for content-based retrieval," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 533–544, 2001.

[29] L. Vuegen, B. Broeck, P. Karsmakers, J. F. Gemmeke, B. Vanrumste,

and H. Hamme, "An MFCC-GMM approach for event detection and classification," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013, pp. 1–3.

[30] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *European Signal Processing Conference (EUSIPCO)*, 2010, pp. 1267–1271.

[31] B. Uzkent, B. D. Barkana, and H. Cevikalp, "Non-speech environmental sound classification using SVMs with a new set of features," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 5, pp. 3511–3524, 2012.

[32] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *International Society for Music Information Retrieval (ISMIR)*, 2016, pp. 805–811.

[33] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 421–425.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[35] Q. Kong, Y. Cao, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems," *arXiv preprint arXiv:1904.03476*, 2019.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations (ICLR)*, 2015.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[39] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning (ICML)*, 2010, pp. 807–814.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[41] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[42] 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.

[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobilenetV2: Inverted residuals and linear bottlenecks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[44] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," in *Sound and Music Computing Conference*, 2017, pp. 220–226.

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

[46] 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.

[47] L. Ford, H. Tang, F. Grondin, and J. Glass, "A deep residual network for large-scale acoustic scene analysis," *INTERSPEECH*, pp. 2568–2572, 2019.

[48] 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, pp. 3107–3111.

[49] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *ACM International Conference on Multimedia*. ACM, 2015, pp. 1015–1018.

[50] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, "Integrating the data augmentation scheme with various classifiers for acoustic scene modeling," *DCASE2019 Challenge, Tech. Rep.*, 2019.

[51] I.-Y. Jeong and H. Lim, "Audio tagging system for DCASE 2018: focusing on label noise data augmentation and its efficient learning," *DCASE Challenge Tech. Rep.*, 2018.

[52] T. Chen and U. Gupta, "Attention-based convolutional neural network for audio event classification with feature transfer learning," *https://cvssp.org/projects/making_sense_of_sounds/site/assets/challenge_abstracts_and_figures/Tianxiang_Chen.pdf*, 2018.

[53] E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, November 2018, pp. 69–73.

[54] C. Kroos, O. Bones, Y. Cao, L. Harris, P. J. Jackson, W. J. Davies, W. Wang, T. J. Cox, and M. D. Plumbley, "Generalisation in environmental sound classification: the 'making sense of sounds' data set and challenge," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 8082–8086.

[55] C. Liu, L. Feng, G. Liu, H. Wang, and S. Liu, "Bottom-up broadcast neural network for music genre classification," *arXiv preprint arXiv:1901.08928*, 2019.

[56] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[57] Y. Zeng, H. Mao, D. Peng, and Z. Yi, "Spectrogram based multi-task audio classification," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3705–3722, 2019.

[58] S. R. Livingstone, K. Peck, and F. A. Russo, "Ravdess: The Ryerson audio-visual database of emotional speech and song," in *Annual Meeting of the Canadian Society for Brain, Behaviour and Cognitive Science (CSBBCS)*, 2012, pp. 1459–1462.