Comparison and Analysis of SampleCNN Architectures for Audio Classification

Taejun Kim, Jongpil Lee, and Juhan Nam, *Member, IEEE*

***Abstract*—End-to-end learning with convolutional neural networks (CNNs) has become a standard approach in image classification. However, in audio classification, CNN-based models that use time-frequency representations as input are still popular. A recently proposed CNN architecture called SampleCNN takes raw waveforms directly and has very small sizes of filters. The architecture has proven to be effective in music classification tasks. In this paper, we scrutinize SampleCNN further by comparing it with spectrogram-based CNN and changing the subsampling operation in three different audio domains: music, speech, and acoustic scene sound. Also, we extend SampleCNN to more advanced versions using components from residual networks and squeeze-and-excitation networks. The results show that the squeeze-and-excitation block is particularly effective among them. Furthermore, we analyze the trained models to provide better understanding of the architectures. First, we visualize hierarchically learned features to see how the filters with small granularity adapt to audio signals from different domains. Second, we observe the squeeze-and-excitation block by plotting the distribution of excitation in several different ways. This analysis shows that the excitation tends to be increasingly class specific with increasing depth but the first layer that takes raw waveforms directly can be highly class specific, particularly in music data. We examine this further and show that the excitation in the first layer is sensitive to the loudness, which is an acoustic characteristic that distinguishes different genres of music.**

***Index Terms*—Audio classification, end-to-end learning, convolutional neural networks, residual networks, squeeze-and-excitation networks, interpretability.**

I. Introduction

CONVOLUTIONAL Neural Networks (CNNs) have proven highly effective in classification tasks where the input is high-dimensional sensory data such as image or audio. In image classification, the end-to-end learning that takes raw pixels of images directly as inputs of the CNNs has become a standard approach [1]–[4]. In audio classification, on the other hand, the majority of CNN-based models still use spectrogram-based inputs such as mel-spectrograms that involve significant hand-crafted design in the time-frequency transform. Thus, depending on the domains in audio classification tasks, researchers have used different (sub-optimal) settings of spectrograms. For example, in music classification tasks, 128 bins of mel-spectrograms is a dominant choice [5]–[9]. In environmental sound classification or speech recognition tasks, 80 bins or less size are more frequently used [10]–[17]. Other than that, the size and type of window, hop size, and the log-compression ratio of the magnitude are also different depending on the domain and specific task. These design choices in turn affect the choice of CNN architecture, e.g., dimensionality of convolutional layer (1D or 2D) and size of filters [7].

The efforts to learn audio representations directly from raw waveforms to uncover hearing mechanism or improve audio classification date back to more than a decade [18]–[21]. However, end-to-end-approach that uses waveforms as input of CNNs was attempted quite recently. Dieleman and Schrauwen applied waveforms to CNNs for music auto-tagging and compared it to mel-spectrogram input [22]. While the patterns of learned filters from waveforms look meaningful, the approach did not outperform mel-filter banks. Sainath *et al.* used Convolutional Long short-term memory Deep Neural Networks (CLDNNs) that take waveforms as input for speech recognition [23]. They showed that raw waveform features match the performance of log-mel filterbank when the network was trained with a large amount of speech data. They noted that a stack of LSTMs presumably helped reducing phase variation in the time-domain convolution. Dai *et al.* used Deep Convolutional Neural Networks (DCNN) with skip connections for environmental sound recognition [24]. They found out that CNNs using waveform input can be competitive with CNNs using mel-spectrogram input when the network layer is sufficiently deep. Other end-to-end approaches for various audio recognition tasks are found in many literatures [25]–[28]. All of these previous works used a large size of filters in the first convolutional layer, which typically takes several hundred samples at a time (or at frame-level). In this case, however, the filters should learn all possible phase variations within the filter size in the time domain. Thus, simply replacing the spectrograms with the corresponding waveforms does not improve performance in plain CNNs [22]. To train CNNs with waveforms properly, very deep networks [24] or carefully designed networks to mitigate the phase variation [23] are necessary along with a large amount of data.

Lee *et al.* recently proposed a different type of CNNs that takes raw waveforms as inputs and has very small sizes of filters [29], [30]. They found the CNNs perform better when the first convolutional layer takes a small grain of samples (e.g., 2 or 3 samples) rather than a typical frame-level size (e.g., 256 or 512 samples)

Manuscript received October 9, 2018; revised February 15, 2019; accepted March 19, 2019. Date of publication April 4, 2019; date of current version May 16, 2019. This work was supported by the National Research Foundation of Korea (Project 2015R1C1A1A02036962). The guest editor coordinating the review of this paper and approving it for publication was Prof. A. Alwan. *(Corresponding author: Juhan Nam.)*

The authors are with the Graduate School of Culture Technology, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: taejun@kaist.ac.kr; richter@kaist.ac.kr; juhannam@kaist.ac.kr).

Digital Object Identifier 10.1109/JSTSP.2019.2909479

in music auto-tagging, and termed the CNN architecture as SampleCNN. Pons *et al.* also showed the effectiveness of this model for music auto-tagging at a scale [31]. This industry-scale experiment showed that the SampleCNN model can outperform spectrogram-based models when the number of songs are sufficient (e.g., more than one millions songs). The authors also conducted a comprehensive evaluation of CNN architectures for (music) audio classification using the property that performances of untrained models (i.e. randomly weighted CNNs) are correlated to performances after training [32], [33]. They showed that SampleCNNs are generally superior to other waveform-based CNNs with large filters. Kim *et al.* improved SampleCNN further for music auto-tagging by investigating more advanced convolutional building blocks developed for image classification [34]. Comparing Residual Networks (ResNets) [2], Squeeze-and-Excitation Networks (SENets) [4] and their combinations, they found that the Squeeze-and-Excitation (SE) blocks from SENets are particularly effective.

We should note that sample-level models that go beyond the frame-level convention were already proposed in the context of audio generation. For example, WaveNet is an auto-regressive generative model that predicts the current sample given a segment of previous samples via dilated convolution layers [35]. SampleRNN is another auto-regressive model using a hierarchical recurrent neural networks [36]. In particular, WaveNet achieved highly impressive results compared to vocoder-based models that use frame-level voice parameters in text-to-speech synthesis. There was also some previous work in waveform-based audio classification using smaller filters than the frame-level size (e.g. 8 samples [37], 10 samples [38], [39], and 32 samples [40]). However, they did not focused on the effect of the small size filters or used only two or three convolution layers which is not sufficient to learn the complex acoustic structures.

In this paper, we investigate SampleCNN and its extended architectures by conducting comprehensive experiments for three different audio classification tasks: music auto-tagging, keyword spotting, and acoustic scene tagging. The experiments focus on scrutinizing the SampleCNN architecture and performance comparison of extended SampleCNN architectures. The results ensure the effectiveness of SampleCNN and show that the SE block is generally most effective for the audio classification tasks. In addition, in order to provide better understanding of SampleCNN and the SE block, we analyze trained models in two ways. First, we visualize the hierarchically learned filters for the three sub-domains of audio data. We show that the six layers in the bottom, which corresponds to approximately one frame of audio, have different distributions of frequency selectivity for the different types of audio. Second, we compute the statistics of excitation in the SE blocks and show how the excitation is associated with the loudness of audio and the activations of feature maps. This confirms that the excitation is associated with discriminating different categories of audio. In particular, excitation in the first layer is sensitive to the loudness which is an acoustic characteristic that distinguishes different genres of music, confirming the finding in [34].

II. SampleCNN Architecture

In this section, we review the architecture of SampleCNN by comparing it to spectrogram-based CNNs and WaveNet.

A. Spectrogram-Based CNNs

Mel-spectrogram is a widely used input representation for audio classification. As shown in Fig. 1, they are obtained via multiple computational steps including Short-Time Fourier Transform (STFT), absolute value operation, linear-to-mel mapping and magnitude compression. This can be in fact seen as a special case of neural networks where affine (or linear) transforms (STFT and linear-to-mel mapping) and non-linearity functions (absolute value operation and magnitude compression) are fixed by hand design [41], [42]. As aforementioned, hyper-parameters of the "fixed networks" such as window size, hop size, mel-filter bank size are usually chosen from the best practice in each task of different audio domains. The 2D time-frequency representation in turn affects the configuration of CNN architecture, for example, the type of convolution (i.e., 1D CNN or 2D CNN) [42] or specific filter sizes that are associated with timbral or temporal characteristics of audio data [7]. In Fig. 1, we set up the CNN model with 1D convolutional blocks to directly compare it to SampleCNN.

B. SampleCNN

SampleCNN is an end-to-end CNN model that takes raw waveforms as input [30], [43]. The architecture is composed of stacks of convolutional layer and subsampling layer. The main difference from other CNN models that take raw waveforms is that it has a very small filter size such as 2 or 3 samples in all layers. This small granularity of filters significantly reduces the possibility of learning the same filter shape at different phases in the time-domain. An example of SampleCNN is shown in Fig. 1 where sub-sampling is done by max-pooling except the first convolutional layer that operates with striding, and the filter size, stride size, and max-pooling size are all 3 samples. This is based on the original SampleCNN architecture proposed in [43]. We can actually reconfigure the architecture by using max-pooling for all layers or more strided convolutions from the bottom. This issue is addressed in the following subsection (Section II-C) in detail.

Fig. 1 indicates that the fixed neural network that computes mel-spectrogram in the spectrogram-based CNN model are comparable to a subset of convolutional blocks from the bottom of the sampleCNN model. For example, if the frame-size is 512 samples in mel-spectrogram, they roughly correspond to 5 blocks of convolutional and pooling layers ($3^5 = 243$ samples). The advantage of sampleCNN is that we can use the learnable blocks for different domains of audio data and they can be optimized to achieve better performance than mel-spectrogram. In Section VI, we will visualize how the learned filters in the blocks are different for music, speech and environmental sounds.

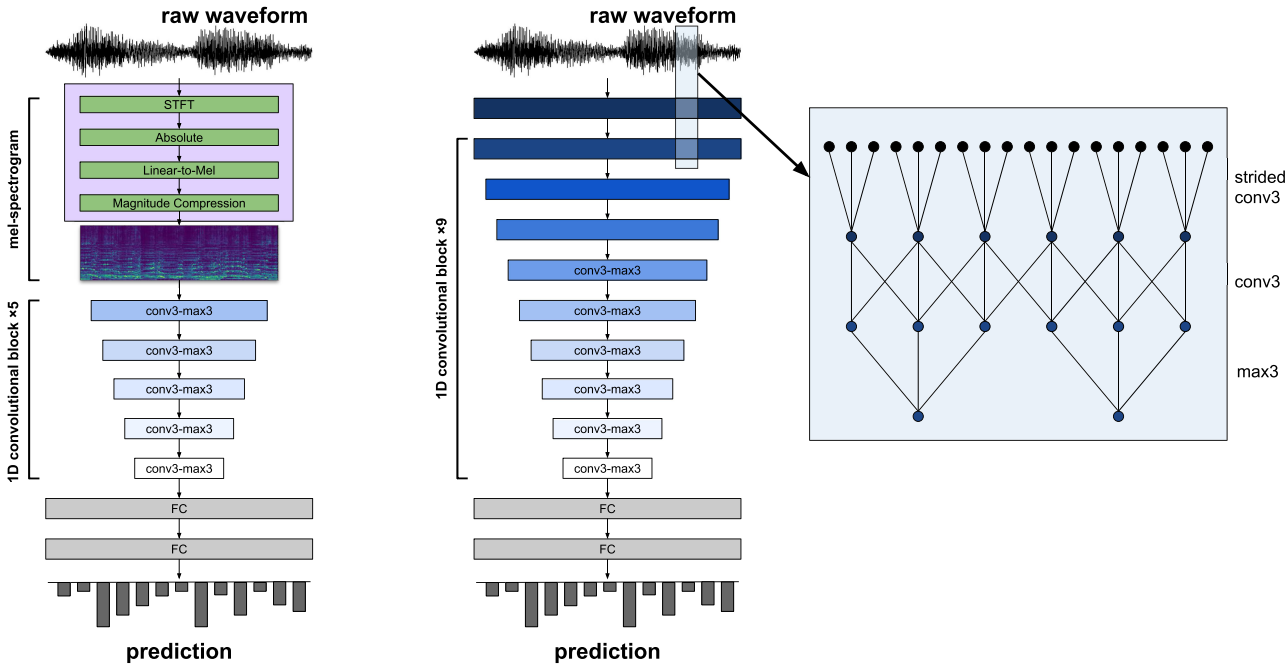


Fig. 1. A CNN model with mel-spectrogram input (left) and a SampleCNN model with raw waveform input (right).

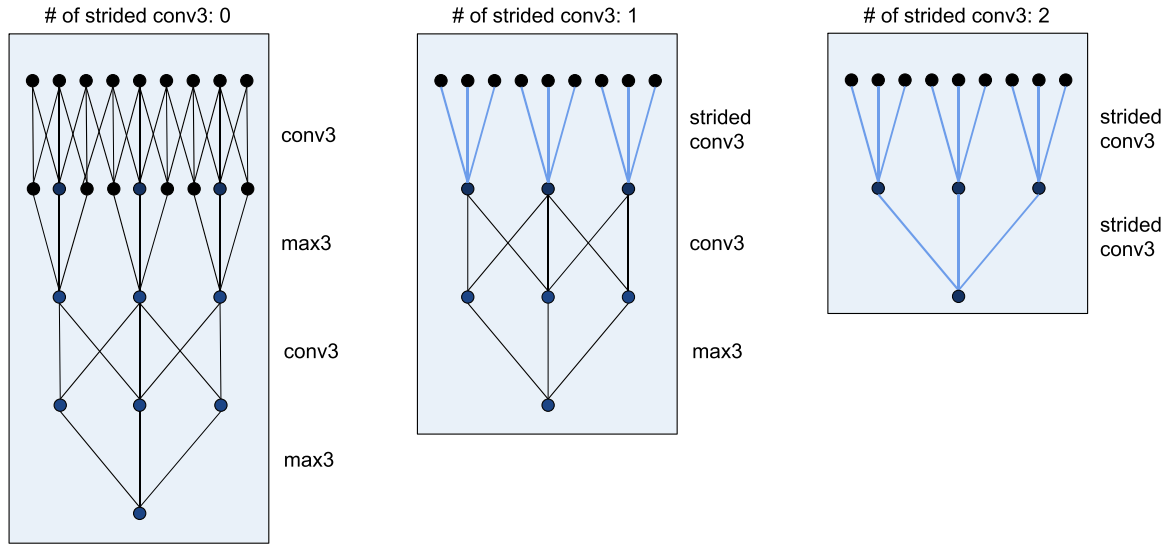
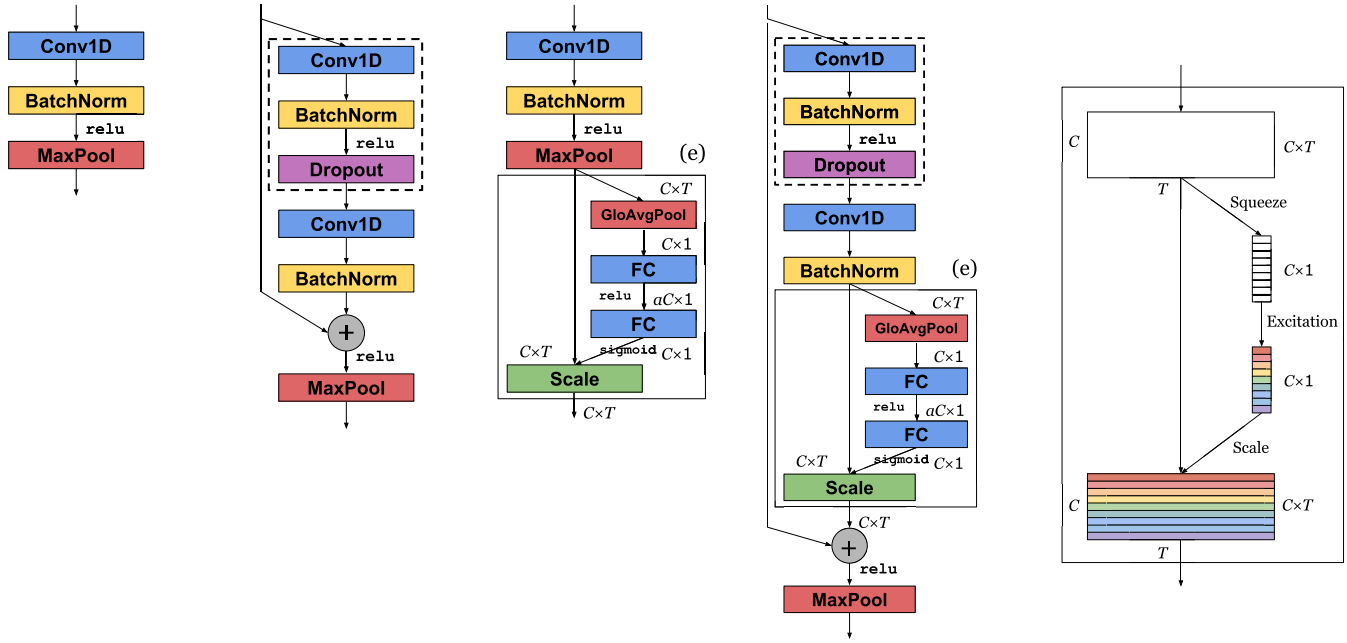


Fig. 2. Toy examples of SampleCNN. The size of receptive field from the top hidden units is 9 in common but the number of strided convolutions changes from 0 to 2. The right one is similar to the Wavenet architecture where both of the filter size and dilation size are 3 samples.

C. Relation With WaveNet

WaveNet is a sample-level deep learning model for generating raw audio waveforms directly [35]. Despite of the different usage, WaveNet and SampleCNN have similar structures in that both of them are based on CNN and the size of receptive field for a hidden unit in the top hidden layer exponentially increases as a factor of n where n is the number of hidden layers. In WaveNet, the exponentially increasing receptive field is implemented by dilated convolutions (typically with a factor of 2), whereas, in SampleCNN, it is done by a pair of convolution and subsampling (either max-pooling or striding) that have the same sizes (typically with a factor of 3 when the sizes are 3). When striding is used for subsampling in SampleCNN, the strided convolution is in fact equivalent to dilated convolution. Figure 2 shows toy examples of SampleCNN where the size of receptive field for the top hidden unit is 9 in common but the number of strided convolutions changes from 0 to 2. When the number of strided convolutions is 2 (the right one), the structure is very similar to WaveNet where the size of dilation is 3, except that SampleCNN is symmetric (or non-causal). Considering this relation between the two architectures, we will examine how the performance of



(a) Basic block (b) Res-n block (c) SE block (d) ReSE-n block (e) Feature recalibration

Fig. 3. 1D convolutional building blocks for audio classification. The basic block is taken from the original sampleCNN in Fig. 1. The Res-n block has a skip connection adopted from ResNets. Res-1 corresponds to the block without the layers in the dotted lines having one convolution layer. Res-2 contains all components having two convolutional layers. SE block is adopted from from SENets. T and C denote dimensionality in time and filter channel, respectively. ReSE-n block is a combination of Res-n and SE blocks. Feature rescaling by the squeeze-and-excitation operations in the shade box of (c) SE block and (d) ReSE-n block is illustrated in (e).

SampleCNN changes when the number of strided convolutions progressively increases from the bottom layer.

III. Extended SampleCNN Architectures

There have been important milestones in image classification centered on the winning models of the ImageNet challenge [44]. They include AlexNet [45], VGGNets [46], ResNets [2], and SENets [4]. The SampleCNN architecture in Fig. 1 is in fact similar to VGGNets with 3×3 filters at each layer except that the filter size, convolution and pooling are one-dimensional. Considering that ResNets and SENets were developed on top of VGGNets, it is natural to plug the components of ResNets or SENets into SampleCNN to make further improvement in audio classification. We previously proposed the extended SampleCNN architectures derived from ResNets or SENet and evaluate them focusing on music auto-tagging [34]. Here we review the architectures in more detail and apply them to speech and acoustic scene data as well with more experiments.

A. Res-n Block

Residual Networks (ResNets) allow to train a very deep CNN using *skip connections* [2]. The shortcuts make backpropagation of gradients more fluent in the neural networks and even enables to train a 1001-layer ResNet [47]. We define the basic block with a single skip connection as *Res-n Block*. n counts the number of convolutional layers and we handle only two cases when n is 1 or 2. Res-2 block is shown in Fig. 3b. It has a dropout layer with a drop ratio of 0.2 before the second convolutional layer to prevent overfitting referring to WideResNets [48]. Res-1 block is the one without the dotted area.

B. Squeeze-and-Excitation (SE) Block

SE block is a new architectural unit that can be added to the basic unit as shown in Fig. 3c. It consists of two operations: *squeeze* and *excitation* operations. The squeeze operation extracts a statistic for each channel, which is called *channel-wise statistic*. It is implemented with a global average pooling layer over time. Thus, $C \times T$ feature map is reduced to $C \times 1$ channel-wise statistics as shown in Fig. 3e. This can be understood as *channel magnitudes* because the outputs from the basic blocks are equal to or greater than zero and the statistics are the averages of filter activations over time for each channel. The excitation operation takes the channel-wise statistics as inputs and calculates a weight for each channel, which is called *excitation*. The weights are learned via the two fully-connected layers. The excitation operation is implemented with two fully-connected (FC) layers. The dimensionality between two FC layers can be adjusted by a hyperparameter α. In our experiment, we did a grid search with $\alpha = [2^{-4}, 2^{-3}, \ldots, 2^4]$. We should note that a small α is preferred because the number of parameters can be significantly increased when α is greater than 1. Each of $C \times 1$ excitations is multiplied to each channel over time, thereby rescaling the feature map as shown in Fig. 3e. This is similar to the gating mechanism applied to each channel separately but it is a lightweight gating to improve the representational power

TABLE I
DESCRIPTION OF THE THREE DATASETS AND MODELS

	Music auto-tagging	**Keyword spotting**	**Acoustic scene tagging**
Dataset	MagnaTagATune [48]	Speech Commands [49]	DCASE17 Task 4 [50]
# of classes	50 tags	35 commands	17 sound events
Labels	Multi-label	Single-label	Multi-label
Sampling rate	22,050Hz	16,000Hz	44,100Hz
Dataset split (train/valid/test)	15,244 / 1529 / 4332	84,843 / 9981 / 11,005	46,042 / 5618 / 1103 (evaluation set)
Duration	29 seconds	1 second	10 seconds
Evaluation metric	ROC-AUC	Accuracy	F-score (instance-based)
Model			
Input size	59,049 samples, 2.68 sec (resampled to 22,050Hz)	22,050 samples, 1 sec (resampled to 22,050Hz)	22,050 samples, 1 sec (resampled to 22,050Hz)
# of segments	10 segments per clip	1 segments per clip	10 segments per clip
# of blocks	9 blocks	8 blocks	8 blocks

of the network by modelling channel-wise relationships [4]. In general, the excitations tend to be increasingly class-specific with increasing depth [4]. However, we will show that the first layer that takes raw waveforms directly can be highly class-specific in music data.

C. ReSE-n Block

ReSE-n block is built by combining the Res-n block and the SE block to take advantages of both blocks. There are multiple possible positions where the SE block can be integrated with the Res-n block [4]. We chose the combination as shown in Fig. 3d as it performs well in our experiments.

IV. TASKS AND DATASETS

One purpose of this study is verifying the universality and the effectiveness of SampleCNN and its extended architectures by evaluating them on the three different audio domains: music, speech and acoustic scene sound. Although there are various audio classification tasks, we choose one for each domain that has been widely studied by researchers. Since the end-to-end learning models generally require sufficient data to train, we also considered the data volume. Table I summarizes the datasets and model settings for each task. We describe each of them in detail below.

A. Music Auto-Tagging

Music auto-tagging is a multi-label classification task. The labels can include genre, mood, instrument, vocal quality and other song-level descriptions. We chose MagnaTagATune (MTT) which is a widely used dataset in the music information retrieval (MIR) community [49]. One of the main evaluation methods in the task is semantic retrieval, which measures the ranking accuracy of retrieved songs for each label. We used area under receiver operating characteristic (ROC-AUC) which is commonly used ranking metric. We calculated the ROC-AUC scores separately for all tags and averaged them to report a single score. We used clips which have at least one positive label and length more than 29.1 seconds.[1] We also evaluated Million Song Dataset (MSD) annotated with the Last.FM tags [52] but used it only for comparing SamplesCNN models with state-of-the-arts in Table IV. We used the same preprocessing techniques for both MTT and MSD. We filtered and split the datasets following previous works [8], [22], [29].

B. Keyword Spotting

Keyword spotting is a multi-class classification task. Instead of continuous conversions where complex speech recognition systems are necessary, this task is designed to support short speech command recognition which is an essential feature in the ever-increasing AI speakers these days. The TensorFlow community recently held a simple speech command recognition challenge and publicly released the speech command dataset.[2] We used the second version of the dataset that includes 35 speech commands [50]. Since this is the problem of categorizing one out of the 35 possible commands, we simple used classification accuracy as an evaluation metric.

C. Acoustic Scene Tagging

Acoustic scene tagging is a multi-label classification task. Specifically, this is defined in the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge 2017 Task 4: Large-scale weakly supervised sound event detection for smart cars.[3] We used only the version without timestamps, which is called "audio tagging". The dataset employed a subset of Google AudioSet.[4] Following the task rule, we used instance-based F-score, which averages F-scores across audio clips in the test set.

D. Model Training

We train all CNN models using stochastic gradient descent with Nesterov momentum of 0.9 and a batch size of 23. The

[1]This filtering reduces the number of songs to about 21K instead of about 26K in the original MTT dataset. This test setup was also conducted in [9].
[2]https://www.kaggle.com/c/tensorflow-speech-recognition-challenge
[3]http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-large-scale-sound-event-detection
[4]https://research.google.com/audioset/

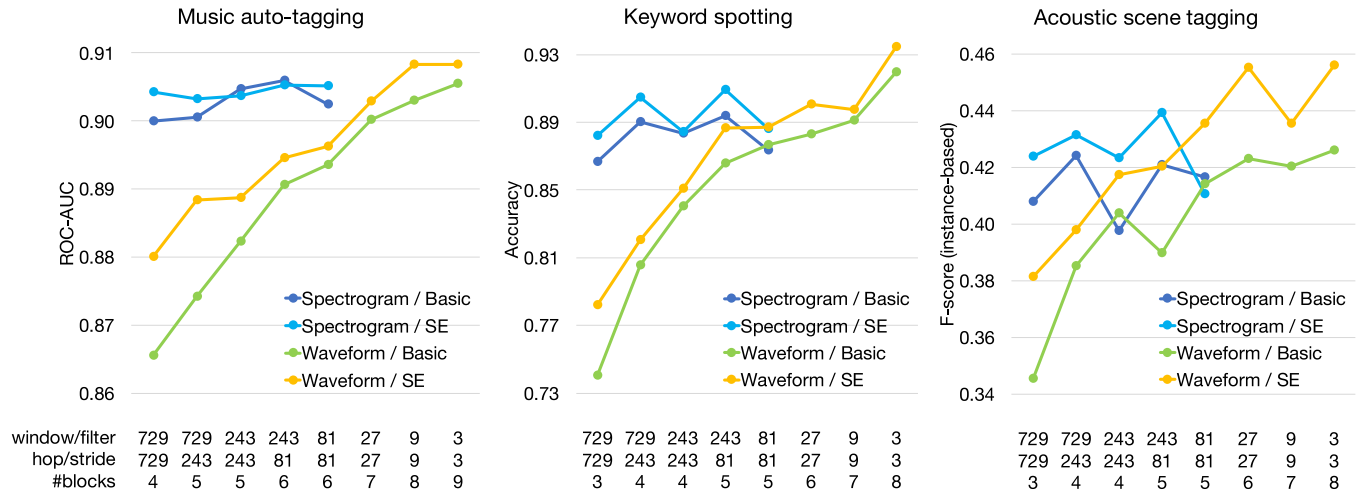


Fig. 4. Performance of spectrogram-based CNNs and SampleCNNs for varying sizes of window (filter) and hop (stride) in the first blocks.

learning rate is initially set to 0.01, and is divided by 5 when the validation loss does not decrease in 2 epochs. A dropout layer of 0.5 was inserted before the last FC layer. In the classification tasks, the prediction was made for each segment on training, whereas the final prediction was for each audio clip by averaging the predictions of the segments. The waveform input was not normalized except in the normalization experiment in Section VII-D. The models were built and trained with TensorFlow[5] and Keras.[6] The source code is available at the link.[7]

V. Performance Comparison

A. SampleCNN

In this subsection, we report experiment results that examine the configuration of SampleCNN.

1) Spectrogram-Based CNNs vs. SampleCNNs: In Section II, we interpreted mel-spectrogram as the output of fixed neural networks where the affine transforms are hand-designed based on domain knowledge. In this perspective, we rigorously evaluate them with varying sizes of window and hop that are hyper-parameters of the networks. For a fair comparison with SampleCNN, we set up two CNNs such that they are as much equivalent to each other as possible. Considering SampleCNN has a size of 3 for both filter and max-pooling/striding, we use the power of 3 instead of the typical power of 2 (e.g., 512 or 1024). The bottom of Fig. 4 shows a set of window and hop sizes of mel-spectrogam, equivalently, filter and hop sizes in the first convolutional layer of SampleCNN while preserving other layers with the configuration in Fig. 1. The first column in Fig. 4 is when the window/filter size is 729 ($=3^6$) and the hop/stride size is 729 as well. On top of that, both of them use 4 convolutional blocks in common. The second column in Fig. 4 reduces hop/stride by a factor of 3. This increases the size of feature map. The max-pooling of the additional convolutional block takes care of it to maintain the same length of input audio. This way, we keep reducing the sizes of window/filter and hop/stride by a factor of 3, and, accordingly, we increase the number of blocks. However, for mel-spectrogram, we stop reducing the window and hop size when they reach 81 samples because they become too short to represent them in the frequency domain. In addition, in order to have a consistent size of mel-spectrograms, we take the same computational procedure regardless of window and hop size. Specifically, we used Hann window, computed 1024-sample FFT after zero-padding, convert the magnitude into 128-bin mel-spectrum and finally compressed it with $\log(1 + C|A|)$ where A is the mel-spectrum and C is the compression strength set to 10. Note that, although this increases the dimensionality in the frequency domain for short windowed samples (e.g. 81 samples), it does not add more information (it is the result of spectral interpolation by zero-padding).

Fig. 4 shows the performance trends on the three audio classification tasks. While spectrogram-based CNNs have saturated results for the varying size of window and hop, SampleCNNs progressively achieve better performance as filter and stride sizes become smaller and the model is deeper. The best performance was obtained when the SampleCNNs have the smallest filter and stride sizes. Also, we evaluate two convolutional blocks, the basic and the SE throughout the experiment. The SE block consistently outperformed the basic block.

As the models become deeper, the number of model parameters also increases in SampleCNN. Thus, the performance gain might be attributed to the larger model size. In order to verify this, we conducted an additional experiment for SampleCNN (with basic blocks) where the number of parameters is fixed by adjusting the number of filters given the model depth such that they consistently have the same number of parameters. Specifically, we fixed it to 1.9M parameters for music auto-tagging and 1.7M for keyword spotting and acoustic scene tagging. Table II shows the results where the experiment settings are the same as those in Fig. 4 except that the number of parameters is fixed. We see that the same increasing trend in this experiment, indicating that the performance gain benefits from the particular structure of SampleCNN.

[5]https://www.tensorflow.org/
[6]https://keras.io/
[7]https://github.com/tae-jun/sampleaudio

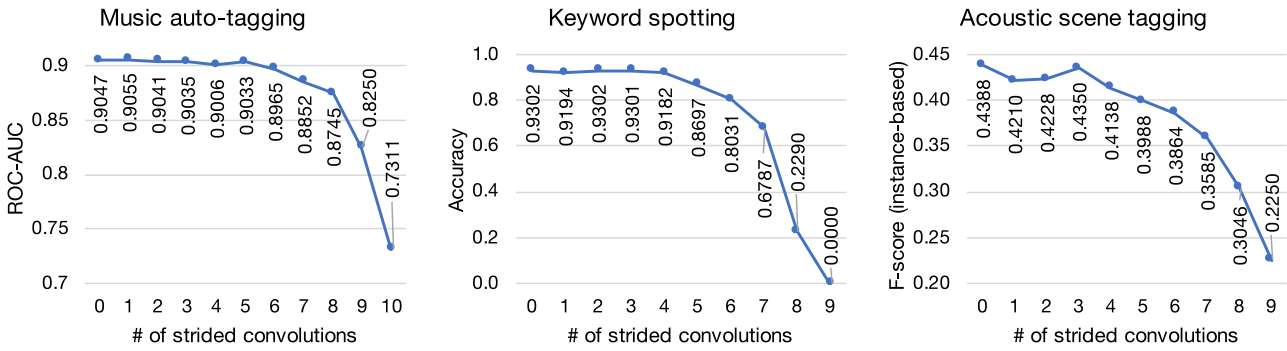


Fig. 5. Performance of SampleCNN (with basic blocks) for the increasing number of strided convolutions from the bottom layer.

TABLE II
EXPERIMENT WITH A FIXED NUMBER OF PARAMETERS (SAMPLECNN WITH BASIC BLOCKS)

Filter size	Stride size	Music auto-tagging	Keyword spotting	Acoustic scene tagging
729	729	0.8684	0.8037	0.3540
729	243	0.8820	0.8668	0.3887
243	243	0.8862	0.8899	0.3921
243	81	0.8945	0.9151	0.3876
81	81	0.8966	0.9146	0.3982
27	27	0.9014	0.9287	0.4237
9	9	0.9045	0.9219	0.4085
3	3	**0.9054**	**0.9253**	**0.4213**

2) Max-Pooling vs. Striding: In Section II-C, we discussed the relation between SampleCNN and WaveNet. They have a common property that the size of receptive field exponentially increases as the model becomes linearly deeper. In particular, when striding is used for subsampling in SampleCNN, the two architectures become more similar to each other. This leads to our curiosity about how the performance will be affected by the number of strided convolutions in SampleCNN. Figure 5 shows the experimental results when the number of strided convolutions increases from the bottom layer of SampleCNN (Note that "0" means that max-pooling is used throughout all layers). They show that the performances are maintained up to half of the depth (4 or 5 strided convolutions) but they are dropped when striding is used more than max-pooling. These results indicate that max-pooling is important in SampleCNN but about half of them can be replaced by striding without losing performance much. That is, half the depth of the model can be replaced by the WaveNet-like structure (Interestingly, the receptive field of half the depth, which is $3^5 = 243$ samples ≈ 11 msec, corresponds to a typical frame size). This is also meaningful in the aspect of computation complexity because striding reduces the number of convolutions 3 times less and saves the max operation.

B. Extended Architectures

In this subsection, we report comprehensive experiment results that compare extended SampleCNN architectures for three audio sets. We set the amplifying ratio α of SE blocks to be 2^{-3} through our experiments since it shows improved performances with the small increment of the number of parameters comparing to basic blocks.

TABLE III
COMPARISON OF COMPUTATIONAL COMPLEXITY (MUSIC AUTO-TAGGING)

Block	# of parameters	Comp. Time	Score (Music)
Basic	1.9M (×1.00)	×1.00	0.9054
SE ($\alpha = 2^{-3}$)	2.0M (×1.09)	×1.08	0.9083
SE ($\alpha = 2^{0}$)	3.3M (×1.74)	×1.14	0.9056
SE ($\alpha = 2^{3}$)	24.0M (×12.80)	×1.14	0.9066
Res-1	2.0M (×1.09)	×1.12	0.9014
Res-2	4.1M (×2.20)	×1.92	0.9070
ReSE-1 ($\alpha = 2^{-3}$)	2.2M (×1.18)	×1.31	0.9038
ReSE-2 ($\alpha = 2^{-3}$)	4.3M (×2.29)	×2.10	0.9075

1) Comparison of the Convolutional Blocks: We evaluated a total of six different versions of SampleCNNs (basic, SE, Res-1, Res-2, ReSE-1, and ReSE-2) across 3 training runs with different initializations for each model and report the mean and standard deviation in the Fig. 6. We also included Precision Recall (PR)-AUC as a supplementary evaluation metric for music auto-tagging task because the primary evaluation metric used in the literature, ROC-AUC, may lead over positive impression when the dataset is unbalanced. From the overall results, we can see that SE, Res-2, and ReSE-2 models are superior to the rest. However, there is no single best model among the three for different audio classification tasks. T-test shows that the differences are statistically non-significant in general. In music auto-tagging (ROC-AUC), the results are as follows: SE $>$ ReSE-2 (p-value $= 0.26$), SE $>$ Res-2 (p-value $= 0.15$), and Res-2 $<$ ReSE-2 (p-value $= 0.56$). In keyword spotting, the results are as follows: SE $<$ ReSE-2 (p-value $= 0.02$), SE $<$ Res-2 (p-value $= 0.04$), and Res-2 $<$ ReSE-2 (p-value $= 0.43$). In the result, Res-2 and ReSE-2 are actually superior to SE. In acoustic scene tagging, we found similar results to those in music auto-tagging. Since the three best models have similar performances, we need other criteria to find a recommended model. This is addressed in the following subsection.

2) Model Complexity: We also report complexity of extended architectures in terms of the number of parameters and computation time in Table III. The computation time was calculated by measuring the elapsed time during one epoch of training. In SE block, the amplifying ratio α significantly increases the number of parameters of SE blocks whereas it does

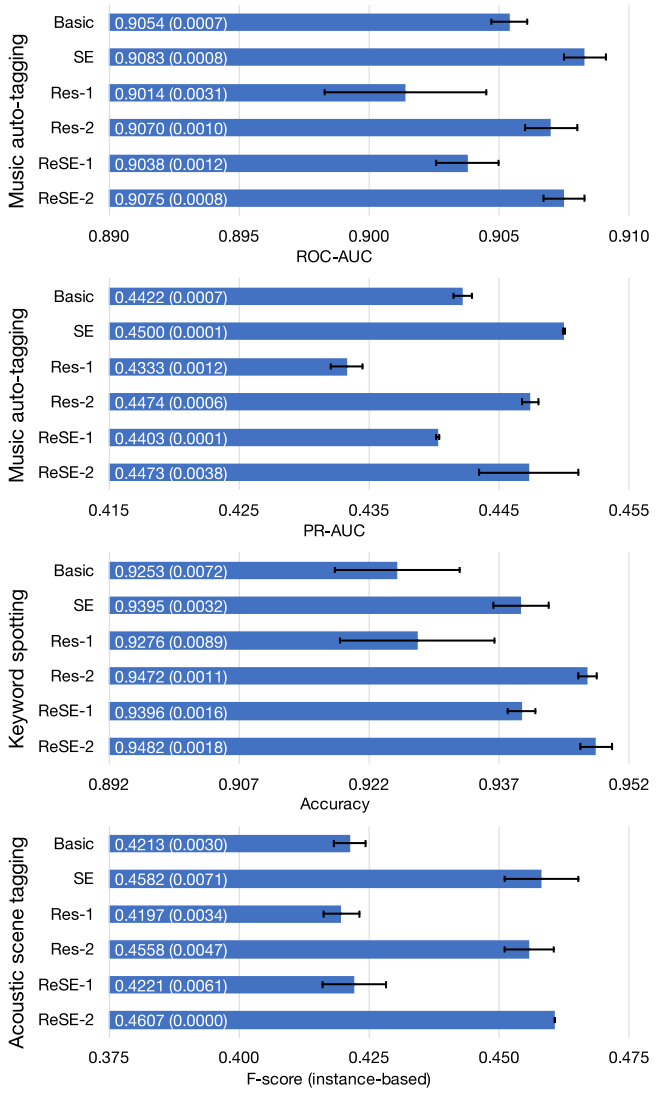


Fig. 6. Performance comparison of convolutional blocks across 3 training runs with different initializations. The numbers indicate means and standard deviations (in parentheses).

not affect computation time much. With $\alpha = 2^{-3}$, the SE block can increase performance well with slightly more parameters and computer time compared to the basic block (therefore, we consistently used $\alpha = 2^{-3}$ throughout the experiments). Res-n blocks require double convolutional layers to gain performance. This results in doubled number of parameters and computation time. ReSE-n also increase performance but even more parameters and computer time. In summary, although SE, Res-2, and ReSE-2 have similar performance, SE is a preferred choice considering the number of parameters and computation time.

C. Comparison With State-of-the-Arts

We also compare our best models with previously reported state-of-the-arts in the three audio classification tasks. We also included music auto-tagging performance on MSD.[8] Table IV summarizes the results. Our best performing architecture for each task is denoted in parentheses. Our SampleCNN models achieved better scores in music auto-tagging and keyword spotting but they are much worse than the state-of-the-art in acoustic scene tagging. We should note that the model denoted with † is from the DCASE challenges and it uses various training techniques such as ensembles, data balancing, and auto-thresholding other than the model itself.

[8]The amplifying ratio of ReSE-2 block is set to be 2^4 in the case of MSD.

TABLE IV
COMPARISON WITH STATE-OF-THE-ARTS

Task	Our Best Score	State-of-the-art
Music auto-tagging (ROC-AUC)		
MagnaTagATune	**0.9083** (SE)	0.9064 [53]
Million Song Dataset	0.8847 (ReSE-2)	**0.8878** [54]
Keyword spotting (Accuracy)	**0.9482** (ReSE-2)	0.9390 [55]
Acoustic scene tagging (F-score)	0.4588 (ReSE-2)	**0.5560**† [56]

VI. Filter Visualizations

Visualizing the filters at each layer allows better understanding of learned features in the hierarchical networks. Following the previous works [29], [30], we used a feature visualization technique called gradient ascent method [57], [58]. It updates the input space such that a particular unit of the networks can be maximized with regards to the input. We targeted the objective function for filter activations at each layer. We averaged the temporal dimension for the filter activations so that the loss can be a single value. Specifically, we first generated random noise of 729 samples and updated it with the gradient of each filter activation while fixing the network parameters. We used 729 samples as input because the length is close to a typical frame size (e.g, 512 or 1024) but it had to be the power of 3. After the inputs are optimized for all filters from the 1st layer (with receptive field of $3^1 = 3$ samples) to the 6th layer (with receptive field of $3^6 = 729$ samples) in the SampleCNN with the basic block, we transform the learned waveforms into log-compressed spectrum using a 729-point FFT. Finally, the spectra are sorted by the frequency at which the magnitude is maximum. The technique is applied to all three audio domains.

The resulting visualizations are displayed in Fig. 7. We can first observe that each filter (a single column) is selective to a small number of frequencies. We sorted them by the frequency at which the magnitude is maximum in the filter. The frequency selectivity curve of learned filters becomes log-like as the layer of networks goes up. This trend is related to the receptive field of the layers. The receptive field of the first layer is only 3 and it is difficult to find such complex nonlinear pattern from the 3 samples. Therefore, the frequency selectivity curve of the first layer approximates to a linear pattern which is similar to the 3-point FFT operation. From the next layer, the receptive field exponentially increase and the filters start to concentrate on low frequency region more, thereby having more log-like curve. However, we should note that the curvatures of the frequency selectivity of learned filters are different among the three audio domains. Interestingly, the learned filter curve of speech

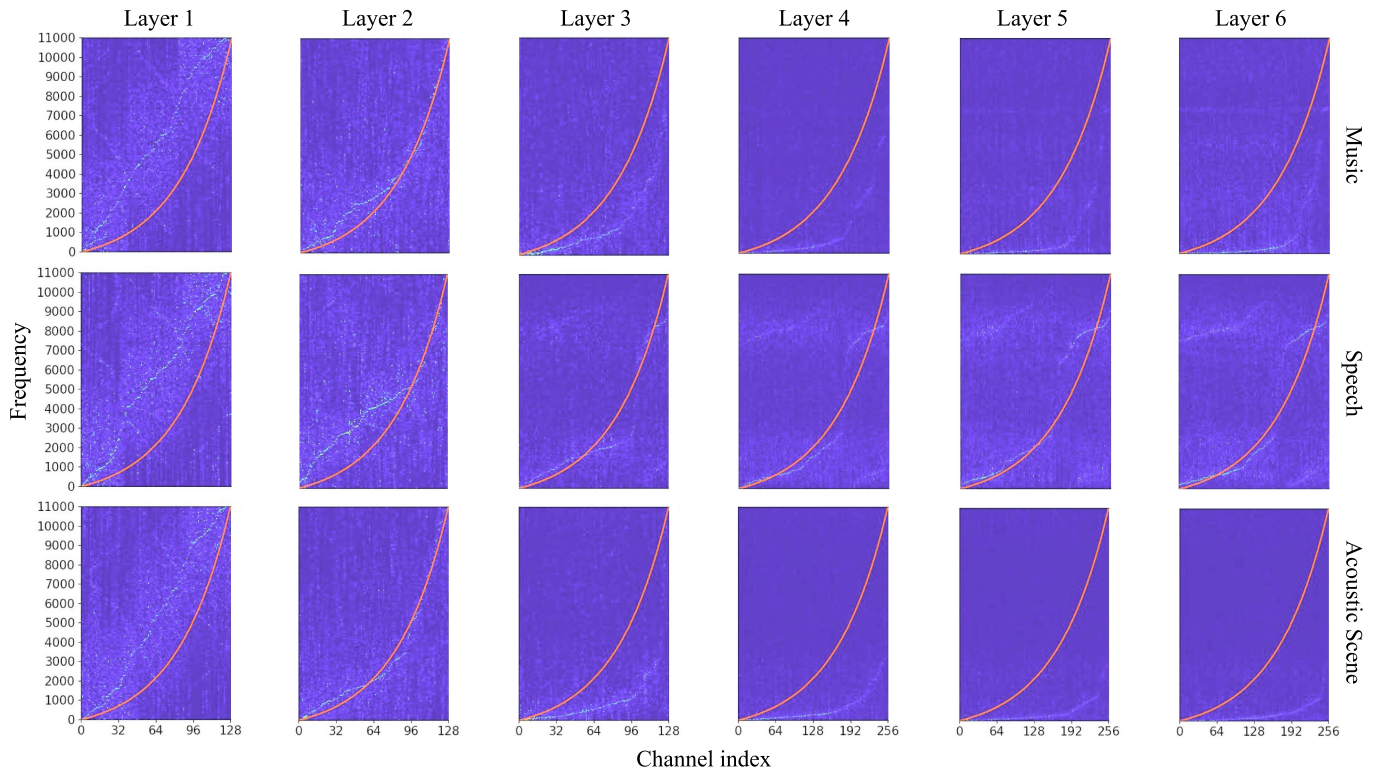


Fig. 7. The spectrum of learned filter estimates for the three datasets in SampleCNN with the basic blocks. They are sorted by the frequency at which the magnitude is maximum. The x-axis represents the index of the filters and the y-axis represents the frequency (ranging from 0 to 11,025 Hz for all figures). The visualizations were obtained using a gradient ascent method that finds the input waveform that maximizes the activation of a filter at each layer. Mel-scale curves are also drawn in red line to compare them with the frequency selectivity curve of learned filters.

sound is very similar to the mel-scale curve when compared to other audio domains. This result makes sense because mel-scale was approximated based on experimental measurements from human subjects. Also, when comparing acoustic scene sound with the other domains, the majority of learned filters concentrate on low-frequency regions. This is probably because audio files in the DCASE task 4 dataset were collected from simple traffic and warning sounds. We assume that this trend reflects the overall spectral distributions in different audio domains and it also represents the adaptation capacity of the waveform-based model.

Meanwhile, considering that the SE block rescales the strength of feature map and the features map is the output of the frequency-selective filters according to the visualization, we could regard the feature rescaling in the SE block as "spectral reweighting".

VII. Excitation Analysis

We showed that the extended SampleCNN architectures generally improve performance in the audio classification tasks. There have been studies that reveal why the skip connections in the ResNets work well. He *et al.* analyzed the skip connections theoretically and showed that they create clean paths for gradients, which lead to the ease of optimization [47]. Li *et al.* explained the effect with visualizations of neural loss functions [59]. Veit *et al.* observed that CNNs with the skip connections exhibit ensemble-like behaviors [60]. However, the behaviors and effects of the SENets have been yet much studied despite of the effectiveness. In this section, we comprehensively analyze the SENets focusing on the relations among audio loudness, channel magnitude and excitation in the context of audio classification tasks. Since we found that analysis results of both SE and ReSE-n blocks are similar, we describe our analysis only for SE blocks.

A. Standard Deviation of Excitations Over Layers

The SE block models channel-wise relationships to improve the representational power of the network. In classification tasks, this is supposed to help discriminating the target labels better and we actually observed the improvement in the previous section. Hu *et al.* showed that the excitations tend to be increasingly class-specific with increasing depth in image classification [4]. That is, the excitations become more uneven for upper layers. We verified this by calculating standard deviations of excitations across all classes at each level of block. Fig. 8 shows the results for the three audio classification tasks. In general, they have increasing trends leading the highest standard deviations near the top level, although the steepness is different for the three datasets. However, in music data, the excitation level in the first block is exceptionally high. We presume that different genres of music have different levels of loudness and the SE block helps discriminating them by detecting the loudness. In the following subsections, we investigate the behaviours of the SE block in the first block more.

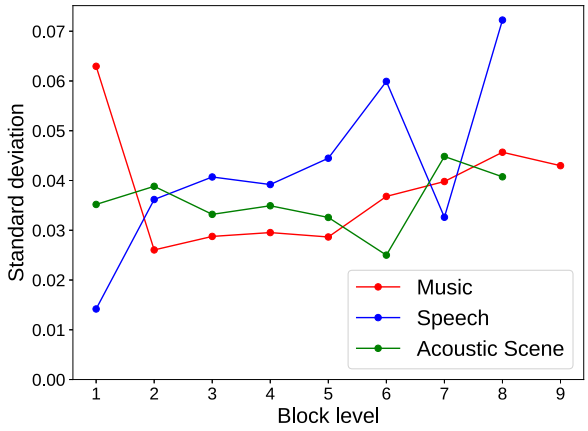


Fig. 8. Standard deviations of excitations across all classes along each layer.

B. Excitations of Distinctive Classes

Observing the excitation for distinctive classes of sounds can provide a better understanding of the SE block. In our previous work [34], we analyzed the average excitation levels of all channels in the first, middle and last blocks for three distinctive genres of music ("classical", "electronic" and "metal"). The result showed that the first block that directly takes the waveform input has the most distinctive distributions for the three genres of music. Specifically, the excitation levels are quite low for metal music whereas they are relatively high for classical music. Considering metal music is generally louder than classical music and the louder input produces higher channel magnitude, we presumed that the SE block in the first layer rescales the activation by detecting the loudness as if audio compressor effect units automatically control the input gain.

In Fig. 9, we plot the excitation levels of the first SE block for all three datasets. The y-axis represents the excitation level and the x-axis represents the channel index. The x-axis is sorted by the average excitation of each channel. We selected three distinctive classes of sounds in each domain by calculating the average Root-Mean-Square (RMS) value for each class and selecting those with low, medium, and high RMS values. The bar plots in Fig. 9 show the relative loudness ratios of the three classes with regards to the highest level. In music, the excitations for "classical", "electronic" and "metal" are highly deviated as seen in the previous work. In speech, soft sounds such "happy" or "bicycle" have relatively high levels of excitation than "forward". However, deviations among them are not strong as the loudness differences are less than those in music. In acoustic scene, "bicycle"and "civil defense siren" are distinctive. As the loudness differences are between music and speech, so do the deviation of excitations.

C. Loudness vs. Excitation and Channel Magnitude

To further manifest the behavior of the SE blocks, we observe the relations among loudness, channel magnitude and excitation for segment-level audio (i.e., the input size of sampleCNN). Fig. 10 shows scatter plots of the channel activation (black dot) and excitation activation (red, blue or green dot) given the loudness of the audio segment for the bottom six SE blocks of the three audio models. Specifically, for a single dot of (x, y), the x-axis value is computed from RMS of the audio segment and the y-axis value is the average of activations over the entire channel for either channel magnitude or excitation.

The distributions of block dots show that the average channel magnitude is strongly correlated with the loudness in the first SE block. That is, as the input audio becomes louder, the average channel magnitude increases. This is not surprising because the channel output of the first block is the result of convolution with the audio input followed by the non-linearity unit. However, the trend becomes diluted as the layer goes up. The distributions of colored dots also show a similar trend but in an opposite direction. That is, as the input audio becomes louder, the average excitation decreases in the first SE block. These results again indicate that the SE blocks normalize the audio in loudness.

TABLE V
NORMALIZATION PERFORMANCE

Task	Block	Normalization	No Normalization
Music auto-tagging	Basic	0.9044±0.0007	**0.9054±0.0007**
	SE	0.9057±0.0005	**0.9083±0.0008**
Keyword spotting	Basic	**0.9318±0.0009**	0.9253±0.0072
	SE	**0.9399±0.0015**	0.9395±0.0032
Acoustic scene tagging	Basic	**0.4426±0.0042**	0.4213±0.0030
	SE	0.4561±0.0041	**0.4582±0.0071**

D. SE Blocks With Waveform Normalization

We showed that SE blocks in the first block normalize input waveforms in the previous sections. However, it is not clear whether the SE blocks simply scale the sound level or do a more complicated processing. To investigate this more, we conducted an additional experiment by explicitly normalizing raw waveforms such that each waveform input for the SampleCNNs has zero mean and unit variance. This normalization (also called standardization) is actually regarded as a standard data preprocessing in training models. Thus, the normalization is likely to help improving the performance. However, we should note that the normalization discards loudness information and thus it can degrade the performance if the target labels are associated with loudness.

Table V shows the mean and standard deviation across 3 training runs with different initializations with or without the segment-level normalization for three audio classification tasks. With basic blocks, the normalization improves the performances in keyword spotting and acoustic scene tagging but not in music auto-tagging. This indicates that keyword spotting and acoustic scene tagging take more benefits from the standardization against losing loudness information whereas music auto-tagging takes more advantage of loudness information. With SE blocks, the normalization does not significantly improve the performances. In both music auto-tagging and acoustic scene tagging,

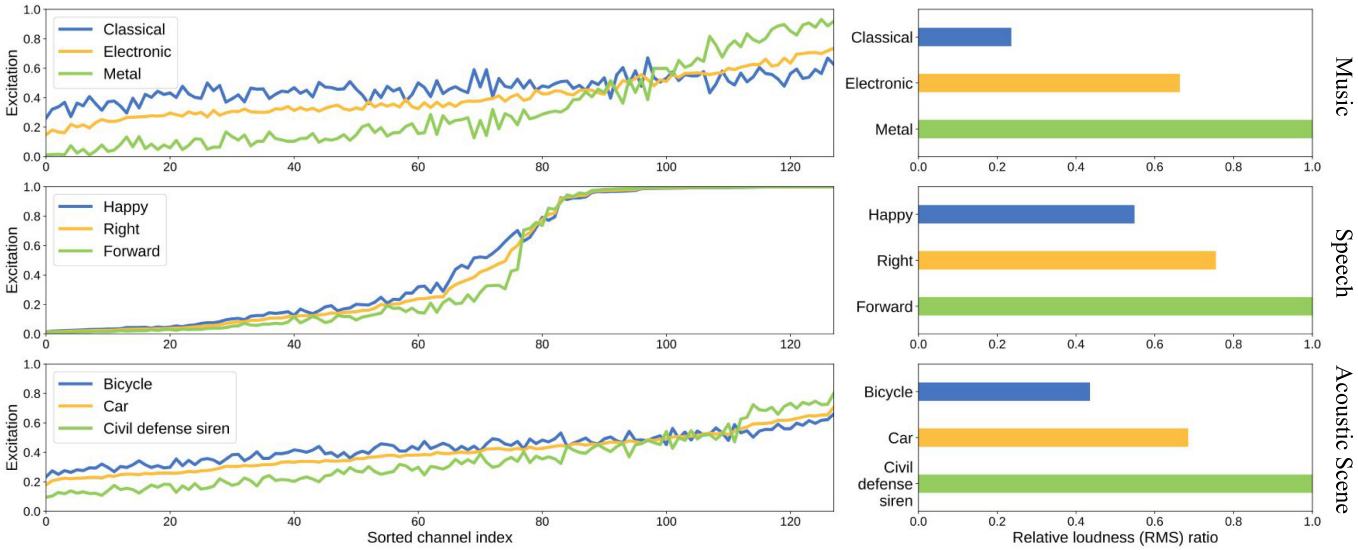


Fig. 9. The excitations from the first level SE blocks (*left*) and relative loudnesses (*right*) of three example classes in each dataset.

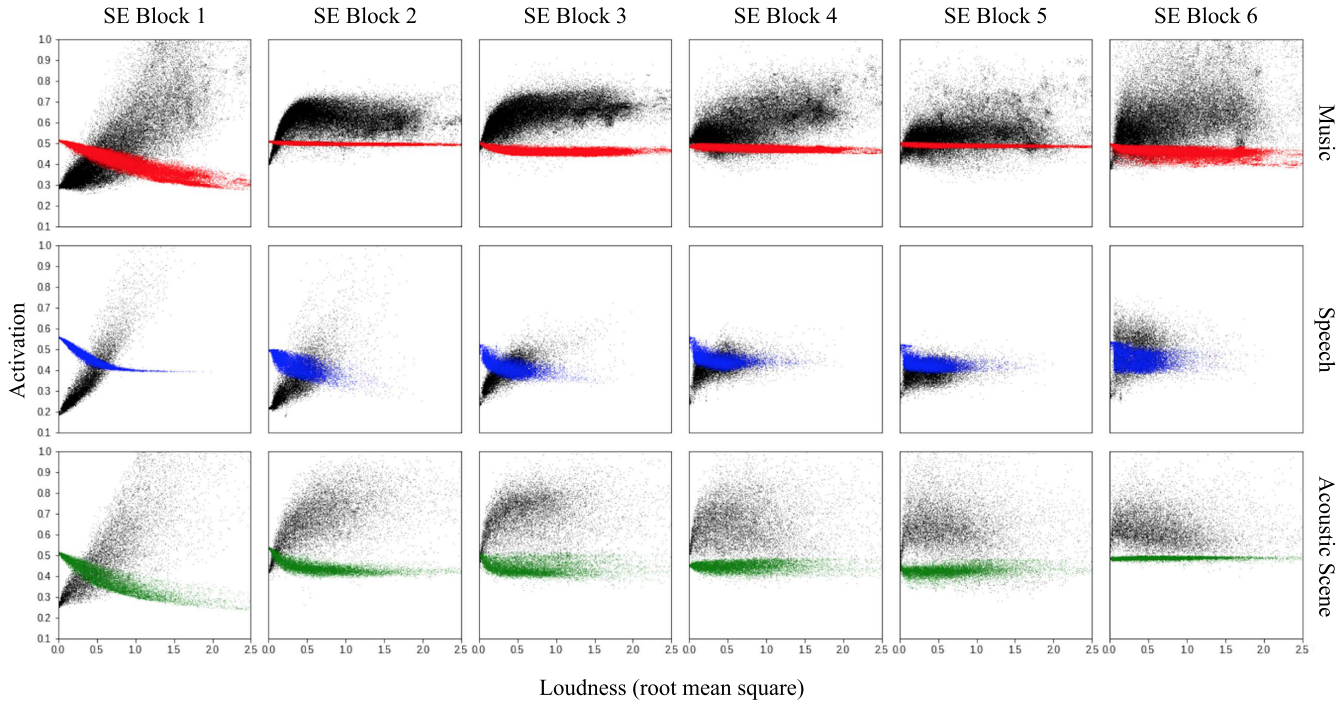


Fig. 10. The scatterplots of loudness vs. activation averaged over all channels at each level of SE block. Each dot represents a segment of an audio clip. Given the loudness of the segment on the x-axis, a*black* dot represents the channel magnitude, and a *colored* dot represents the excitation obtained from each dataset (*red*: music, *blue*: speech, *green*: acoustic scene).

the scores become worse after the normalization. In keyword spotting, the difference is not significant. This can be explained by the effect of scaling the loudness in the SE blocks. That is, since the SE blocks already deal with the sound level, the normalization does not help. However, the SE blocks consistently outperform the basic blocks with the normalized input in keyword spotting and acoustic scene tagging where the loudness information is less important. This indicates that the SE blocks conduct a more sophisticated processing beyond simply scaling the sound level.

We investigated the input normalization in another perspective by measuring the performance difference by the normalization for each tag in the music auto-tagging task. Fig. 11 shows the performance differences of ROC-AUC scores of models with SE blocks when the input waveforms are normalized or not. The difference is calculated by subtracting scores with the

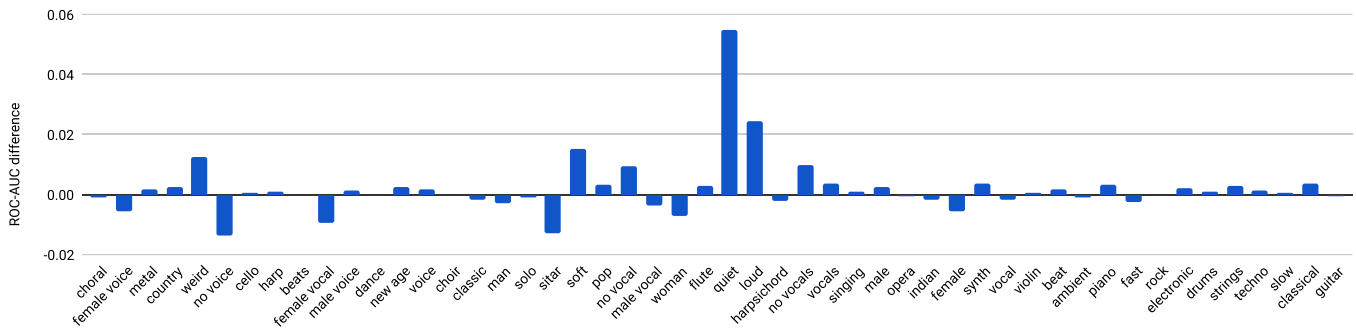


Fig. 11. Performance differences of tags when the segment-level input normalization is applied or not in SampleCNN with SE blocks. The positive bars indicate that the model without the normalization is better.

normalization from scores without it. Therefore, the positive bar means that the performance is better without normalization, and vice versa. Interestingly, "quiet", "loud" and "soft", which are directly related to loudness, have highest positive differences. This ensures that loudness is important information in music auto-tagging and SE blocks prefer unnormalized input.

VIII. Conclusion

In this paper, we examined SampleCNN and its extended architectures by conducting comprehensive experiments for three different audio datasets. Delving into the architecture of the sample-level CNN with comparison to spectrogram CNN and WaveNet, we verified that the structure with very small sizes of filters is effective and max-pooling can be replaced with striding up to half the blocks from the bottom without degrading performance much. Among the extended SampleCNN architectures, SE block was most effective considering both performance and computational efficiency, indicating that the channel-wise rescaling of feature maps improves discriminative power in audio classification tasks. To understand SampleCNN better, we visualized filters for the three SampleCNN models trained with different domains of audio. The frequency selectivity curves of learned filters become more log-like as the layer goes up. In particular, the frequency selectivity curves from speech data is similar to mel-scale. Through the analysis of the SE blocks, we observed that the excitation generally becomes more discriminative in upper layers but the first layer for music data is exceptional as the loudness is important information to distinguish different types or music. We also examined the relationship between SE block and the waveform normalization. This shows that the SE block conducts a sophisticated processing beyond simple scaling of the sound level.

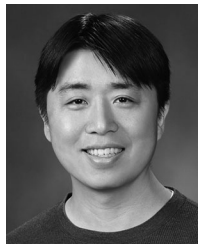
References

[1] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1–9.
[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.
[3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 4700–4708.
[4] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 7132–7141.
[5] S. Dieleman and B. Schrauwen, "Multiscale approaches to music audio feature learning," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2013, pp. 116–121.
[6] A. Van Den Oord, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 29–34.
[7] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *Proc. Content-Based Multimedia Indexing Workshop*, 2016, pp. 1–6.
[8] K. Choi, G. Fazekas, and M. B. Sandler, "Automatic tagging using deep convolutional neural networks," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 805–811.
[9] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proc. Int. Conf. Acoust., Speech. Signal Process.*, 2017, pp. 2392–2396.
[10] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, Mar. 2017.
[11] S. Hershey *et al.*, "CNN architectures for large-scale audio classification," in *Proc. Int. Conf. Acoust., Speech. Signal Process.*, 2017, pp. 131–135.
[12] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *Proc. Int. Conf. Acoust., Speech. Signal Process.*, 2018, pp. 121–125.
[13] Y. Sakashita and M. Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," in *Proc. Detection Classification Acoust. Scenes Events*, Sep. 2018.
[14] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.
[15] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 577–585.
[16] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," in *Proc. Int. Conf. Acoust., Speech. Signal Process.*, 2018, pp. 5934–5938.
[17] J. Guo, N. Xu, L.-J. Li, and A. Alwan, "Attention based CLDNNS for short-duration acoustic scene classification," in *Proc. Int. Speech Commun. Assoc.*, 2017, pp. 469–473.
[18] M. S. Lewicki, "Efficient coding of natural sounds," *Nature Neurosci.*, vol. 5, no. 4, pp. 356–363, 2002.
[19] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 23, pp. 978–982, 2006.
[20] P.-A. Manzagol, T. Bertin-Mahieux, and D. Eck, "On the use of sparse time-relative auditory codes for music," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2008, pp. 603–608.
[21] N. Jaitly and G. Hinton, "Learning a better representation of speech soundwaves using restricted Boltzmann machines," in *Proc. Int. Conf. Acoust., Speech. Signal Process.*, 2011, pp. 5884–5887.
[22] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 6964–6968.
[23] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNS," in *Proc. Int. Speech Commun. Assoc.*, 2015, pp. 1–5.

[24] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 421–425.
[25] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning features of music from scratch," in *Proc. Int. Conf. Learn. Representations*, 2017.
[26] Z. Zhu, J. H. Engel, and A. Hannun, "Learning multiscale features directly from waveforms," in *Proc. Int. Speech Commun. Assoc.*, 2016, pp. 1305–1309.
[27] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," 2016, arXiv:1609.03193.
[28] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur, "Acoustic modelling from the signal domain using CNNs," in *Proc. Int. Speech Commun. Assoc.*, 2016, pp. 3434–3438.
[29] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," in *Proc. Sound Music Comput. Conf.*, 2017, pp. 220–226.
[30] J. Lee, J. Park, K. L. Kim, and J. Nam, "SampleCNN: End-to-end deep convolutional neural networks using very small filters for music classification," *Appl. Sci.*, vol. 8, no. 1, 2018, Art. no. 150.
[31] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 637–644.
[32] A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng, "On random weights and unsupervised feature learning," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1089–1096.
[33] J. Pons and X. Serra, "Randomly weighted CNNS for (music) audio classification," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2019.
[34] T. Kim, J. Lee, and J. Nam, "Sample-level CNN architectures for music auto-tagging using raw waveforms," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 366–370.
[35] A. van den Oord *et al.*, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*. [Online]. Available: https://arxiv.org/abs/1609.03499
[36] S. Mehri *et al.*, "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proc. Int. Conf. Learn. Represent.*, 2017.
[37] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 2721–2725.
[38] D. Palaz, M. Magimai-Doss, and R. Collobert, "Analysis of CNN-based speech recognition system using raw speech as input," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 11–15.
[39] D. Palaz, M. M. Doss, and R. Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4295–4299.
[40] J. Guo, N. Xu, X. Chen, Y. Shi, K. Xu, and A. Alwan, "Filter sampling and combination CNN (FSC-CNN): A compact CNN model for small-footprint ASR acoustic modeling using raw waveforms," in *Proc. Int. Speech Commun. Assoc.*, 2018, pp. 3713–3717.
[41] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Feature learning and deep architectures: new directions for music informatics," *J. Intell. Inf. Syst.*, vol. 41, pp. 461–481, 2013.
[42] J. Nam, K. Choi, J. Lee, S.-Y. Chou, and Y.-H. Yang, "Deep learning for audio-based music classification and tagging: Teaching computers to distinguish rock from Bach," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 41–51, Jan. 2019.
[43] J. Lee, T. Kim, J. Park, and J. Nam, "Raw waveform-based audio classification using sample-level CNN architectures," in *Proc. Workshop Mach. Learn. Audio Signal Process. NIPS*, 2017.
[44] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[46] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 730–734.
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 630–645.
[48] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vision Conf.*, 2016, pp. 87.1–87.12.
[49] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2009, pp. 387–392.
[50] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*.
[51] A. Mesaros *et al.*, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Proc. Detection Classif. Acoust. Scenes Events Workshop*, Nov. 2017, pp. 85–92.
[52] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2011, vol. 2, paper 10.
[53] J. Lee and J. Nam, "Multi-level and multi-scale feature aggregation using sample-level deep convolutional neural networks for music classification," in *Proc. Mach. Learning Music Discovery Workshop/Int. Conf. Mach. Learn.*, 2017.
[54] J. Lee and J. Nam, "Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1208–1212, Aug. 2017.
[55] D. C. de Andrade, S. Leo, M. L. D. S. Viana, and C. Bernkopf, "A neural attention model for speech command recognition," 2018, *arXiv:1808.08929*.
[56] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Surrey-CVSSP system for DCASE2017 challenge task4," in *Proc. Detection Classif. Acoust. Scenes Events Challenge*, 2017.
[57] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," Univ. Montreal, Montreal, QC, Canada, Tech. Rep. 1341, 2009.
[58] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," Distill, 2017. [Online]. Available: https://distill.pub/2017/feature-visualization
[59] H. Li, Z. Xu, G. Taylor, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6389–6399.
[60] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 550–558.

Taejun Kim received the B.S. and M.S. degrees from the School of Electrical and Computer Engineering, University of Seoul, Seoul, South Korea. He is currently working toward the Ph.D. degree with the Graduate School of Culture Technology, Korea Advanced Institute of Science and Technology, Daejeon, South Korea. His current research interests include machine learning for music and audio signal processing for music applications.

Jongpil Lee received the B.S. degree in electrical engineering from Hanyang University, Seoul, South Korea, in 2015, the M.S. degree, in 2017, from the Graduate School of Culture Technology, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, where he is currently working toward the Ph.D. degree. From July to September 2017, he was an intern with Naver Clova Artificial Intelligence Research. His current research interests include machine learning and signal processing applied to audio and music applications.



Juhan Nam (M'09) received the Ph.D. degree in music from the Center for Computer Research in Music and Acoustics, Stanford University, Stanford, CA, USA, in 2013. He is an Assistant Professor with the Graduate School of Culture Technology, Korea Advanced Institute of Science and Technology, Daejeon, South Korea. He was a Staff Research Engineer with Qualcomm, San Diego, CA, USA, from 2012 to 2014. He is interested in various topics at the intersection of music, audio signal processing, and machine learning.