Siamese Style Convolutional Neural Networks for Sound Search by Vocal Imitation

Yichi Zhang, *Student Member, IEEE,* Bryan Pardo, *Member, IEEE,* and Zhiyao Duan, *Member, IEEE*

***Abstract*—Conventional methods for finding audio in databases typically search text labels, rather than the audio itself. This can be problematic as labels may be missing, irrelevant to the audio content, or not known by users. Query by vocal imitation lets users query using vocal imitations instead. To do so, appropriate audio feature representations and effective similarity measures of imitations and original sounds must be developed. In this paper, we build upon our preliminary work to propose Siamese Style Convolutional Neural Networks (SS-CNN) to learn feature representations and similarity measures in a unified end-to-end training framework. Our Siamese architecture uses two CNNs to extract features, one from vocal imitations and the other from original sounds. The encoded features are then concatenated and fed into a fully connected network to estimate their similarity. We propose two versions of the system: IMINET is symmetric where the two encoders have an identical structure and are trained from scratch, while TL-IMINET is asymmetric and adopts the transfer learning idea by pre-training the two encoders from other relevant tasks: spoken language recognition for the imitation encoder and environmental sound classification for the original sound encoder. Experimental results show that both versions of the proposed system outperform a state-of-the-art system for sound search by vocal imitation, and the performance can be further improved when they are fused with the state of the art system. Results also show that transfer learning significantly improves the retrieval performance. This paper also provides insights to the proposed networks by visualizing and sonifying input patterns that maximize the activation of certain neurons in different layers.**

***Index Terms*—Vocal imitation, information retrieval, Siamese style convolutional neural networks, transfer learning, metric learning.**

I. Introduction

DESIGNING ways to efficiently access multimedia documents, such as audio recordings, is an important information retrieval task. The standard approach to index and search audio documents is based on text metadata and conventional text search engines. There are, however, many scenarios where this approach has limited utility. In online repositories, like freesound.org, the metadata often does not describe the relevant details of the audio content, making the target file undiscoverable or submerged within hundreds of results returned by text-based queries.

Even for well-organized sound effect libraries with an explicit hierarchical ontology, searching for sounds is still not easy. It requires the user to be familiar with the taxonomy and text descriptors. In many situations, however, this requirement is very difficult to satisfy, as many sounds, such as computer synthesized sound effects, do not have accurate and commonly accepted text descriptors.

A query-by-vocal-imitation sound retrieval system addresses these issues [1], [2]. Such a system takes a vocal imitation from the user as a query, and searches for sounds in the library similar to the query. This approach does not have to be done in isolation. It can be combined with text-based queries, providing more effective and accurate results. This approach is especially useful for sound retrieval in large-scale libraries where many different sounds share the same text label and in long recordings where the temporal location of labeled events is not known.

Vocal imitation is a common human behavior that uses the vocal organs to mimic sounds. It is an effective way to convey ideas that are difficult to describe in words. For example, callers to the National Public Radios Car Talk show [3] would call in and illustrate symptoms of their vehicles by imitating the sounds caused by mechanical or electrical failures. These imitations make the conversations more effective. Designing computer systems that can recognize vocal imitation for sound search [4], [1] has broad applications. These include music production, for the search of sound effects, security and surveillance for the identification of sound events in long recordings, and biodiversity monitoring for the recognition of bird species in the field.

There are two main challenges in designing vocal-imitation-based sound retrieval systems: feature representation and matching algorithms. Feature representations of vocal imitations and their corresponding original sounds should emphasize the aspects that humans use to imitate sounds. They also need to downplay differences between imitations and original sounds, caused by the physical constraints of the human vocal system. The matching algorithm needs to work with the feature representations to discern target sounds from irrelevant sounds, given a query.

In this paper, we address the two challenges in a unified framework. We do this by extending our previous work on Convolutional Siamese Network [5], [6] to a more general Siamese Style Convolutional Neural Network (SS-CNN). A Siamese network contains two encoders with identical structures to encode two inputs [7]. The proposed Siamese style network contains two similar encoders whose structures can be varied from each other to suit each encoder's respective input. In our system, the two inputs are a vocal imitation query and an original sound from the database to be searched.

Y. Zhang and Z. Duan are with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, 14627 USA e-mail: (yichi.zhang@rochester.edu, zhiyao.duan@rochester.edu).

B. Pardo is with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, 60208 USA e-mail: (pardo@northwestern.edu).

Manuscript received XX XX, XXXX; revised XX XX, XXXX.

Each of the two encoders uses a CNN to extract features from the audio spectrograms. These features are then concatenated and fed into a Fully Connected Network (FCN) to calculate the similarity between the imitation and the sound candidate. The CNN encoders and the FCN are trained simultaneously using positive pairs (an original sound and a vocal imitation of that sound) and negative pairs (the vocalization of a sound different than the one it is paired with). Feature learning and the matching algorithm are thus jointly optimized.

We present two versions of the proposed SS-CNN system. The first one is a symmetric model presented in our preliminary work named IMINET [5], where the two encoders share exactly the same structure, although the weights of the encoders may be different. The entire network is trained directly on positive and negative training pairs of imitations and sound candidates from scratch. The second one is a less symmetric model, where the two encoders use different and domain-specific structures. The imitation encoder uses a structure previously applied to a spoken language recognition task. The original sound encoder adopts a structure used for an environmental sound classification task. Moreover, the encoders are pre-trained separately on these tasks before being fine-tuned on the positive and negative pairs of imitations and sound candidates. Since this lets learned concepts be transferred from other tasks, this version is named TL-IMINET, noting its Transfer Learning (TL) nature.

Experiments are conducted on the VocalSketch Data Set v1.0.4 [8]. Results show that the proposed Siamese style networks outperform state-of-the-art systems [9], [10] where feature learning and matching algorithms are optimized separately. Results also show that transfer learning significantly improves the system performance. To provide insights to the proposed networks, we visualize and sonify input patterns that maximally excite certain neurons and filters.

The main contributions of this work are threefold. First, building upon our preliminary work [5], [6], we propose a novel network architecture (SS-CNN) for sound retrieval by vocal imitation. This architecture jointly optimizes feature learning and metric learning in an end-to-end training framework. Second, we design the TL-IMINET version of the proposed system leveraging the idea of transfer learning from other audio tasks (language classification and urban sound classification). We show that the CNN encoders that are pretrained on these audio tasks can extract effective audio representations for vocal imitations and original sounds, leading to significantly better sound retrieval performance. Third, we visualize and sonify input patterns that excite the learned filters/neurons in different layers of our proposed neural network to provide insights on what the networks are learning in the query-by-vocal-imitation task.

The rest of the paper is organized as follows: We first review related work in Section II, then introduce the Siamese style neural network structure in Section III. In Sections IV and V, we describe the proposed IMINET and TL-IMINET versions in detail. Section VI compares the performance of the proposed systems against state-of-the-art systems. Section VII provides insights to the proposed networks through visualization and sonification of input patterns that excite certain neurons and filters. Finally Section VIII concludes the paper.

II. Related Work

Query by vocal imitation falls into the task of Query by Example (QBE) [11]. There are numerous QBE applications in the audio domain, such as cover song recognition [12] and spoken document retrieval [13]. Audio fingerprinting [14] is also a type of QBE. Originally, it required a portion of the target audio file as the query. Recently, it has been extended to include finding live versions of a song whose studio recording is in the database [15], [16]. Vocal imitation of a sound takes this one step further, and has been shown to be useful in many scenarios, such as finding songs by humming the melody as a query [17], [18] or beat boxing the rhythm [19], [20]. However, little work has been reported on general sound search by vocal imitation.

Roma and Serra [21] designed a QBE system that allows users to search sounds on freesound.org by capturing audio with a microphone as the query. Handcrafted features like statistics of Mel-Frequency Cepstral Coefficients (MFCC) and their derivatives were adopted as descriptors for a given audio clip, but no formal evaluation was reported. Blancas *et al.* [4] built a closed-set supervised system for sound query by vocal imitation using hand-crafted features extracted by the Timbre Toolbox [22] and an SVM classifier. A vocal imitation query was classified to a pre-defined class and sounds in that class were retrieved. The major limitation of closed-set supervised systems, however, is that they cannot retrieve sounds that do not have training imitations. Helén and Virtanen [2] designed a query by recordings system for sound effects. Hand-crafted frame-level features were extracted from both query and sound samples and the query-sample pairwise similarity was measured on probability distributions of the features.

In our own prior work [9], [10], we proposed a system for sound search by vocal imitation called IMISOUND. We employed a Stacked Auto-Encoder (SAE) to learn feature representations from vocal imitations of sounds not contained in the search database and applied this same representation to both queries and sound recordings during the search. We then calculated their similarity through Kullback-Leibler (K-L) divergence [23], Dynamic Time Warping (DTW) [24], and cosine similarity. The feature representation and matching algorithm in IMISOUND, however, were designed separately. This means that the learned features may not be optimal for the similarity measure.

Siamese networks were first proposed by Bromley et al. [7] for signature verification. Since then, they have been successfully applied to many image/video tasks such as face verification [25] and image recognition [26]. More recently, Bertinetto et al. [27] proposed a fully-convolutional Siamese network for object tracking in videos. Han et al. [28] proposed the MatchNet for patch-level image matching, a two tower structure with convolutional layers for feature extraction and fully connected layers for metric learning. Chen and Salman [29] proposed a regularized Siamese deep network to extract speaker-specific information from MFCCs for a speaker recognition task.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2868428, IEEE/ACM Transactions on Audio, Speech, and Language Processing

To overcome the problem of separately learning the distance metric and feature representations, we developed a preliminary model named IMINET [5] that uses a semi-Siamese architecture to calculate the similarity between an imitation query and a sound in the search database. IMINET uses two CNN towers to extract features from the two inputs. The features are then concatenated and fed into a Fully Connected Network (FCN) to calculate their similarity. The CNN encoders and the FCN are trained simultaneously using positive and negative pairs of imitations and sound candidates, leading to superior results over IMISOUND.

To understand what a neural network learns, several visualization methods have been developed [30]. The most straightforward method is to visualize the activations of each layer [31]. Another method is activation maximization [32], which generates an input that maximally activates a certain neuron by performing gradient ascent of the neuron activation w.r.t. the input while keeping the filter fixed. A related technique is to search for the inputs within a dataset that maximally activate a neuron [33]; this requires a large dataset including extensive input patterns. In [34], Deconvnet is proposed to interpret the function of intermediate convolutional layers, where the hidden layer activations are mapped back to the input pixel space using deconvolution and unpooling.

In this paper, we extend this "semi-Siamese architecture" idea by relaxing the structural symmetry normally used in Siamese networks to explore non-symmetric Siamese-style structures. We also compare alternative network structures and late fusion techniques. We then alleviate the data scarcity issue by applying a kind of transfer learning. We use domain-specific encoder architectures that are pre-trained on different, but relevant, datasets and tasks. Finally, we visualize and sonify input patterns that activate the neurons in different layers of our proposed model to provide insights on what the networks are learning in the query-by-vocal-imitation task. The sum of this greatly extended our preliminary work.

III. Proposed Siamese Style Convolutional Neural Networks

As shown in Figure 1, our proposed Siamese Style Convolutional Neural Networks (SS-CNN) can be represented by a generic model that contains two Convolutional Neural Network (CNN) towers for feature extraction: One tower receives a vocal imitation (the query) as its input. The other receives a sound from the searchable database (the candidate) as its input. Each tower outputs a set of features (also known as an embedding). These features are then concatenated and fed into a Fully Connected Network (FCN) for similarity calculation. The final similarity output of the Siamese-style neural network is the probability of being a positive pair between the query and the candidate. The CNNs and the FCN are trained jointly on positive (i.e., related) and negative (i.e., non-related) query-candidate pairs. Through this joint optimization, feature representations learned by the CNNs are better tuned for the FCN's metric learning.

Once the Siamese-style neural network is trained, it can be used to search for sounds in a database of audio files, using a vocal imitation as the query. To do so, we pair the imitation query with each sound candidate in the database and use the neural network to calculate the similarity, where a sound candidate refers to the original sound recording representing a certain concept within the entire dataset. Let the similarity for the i-th sound candidate be $p_{ssn}(i)$. We then rank all sound candidates by their probabilities from high to low and return them in this order.

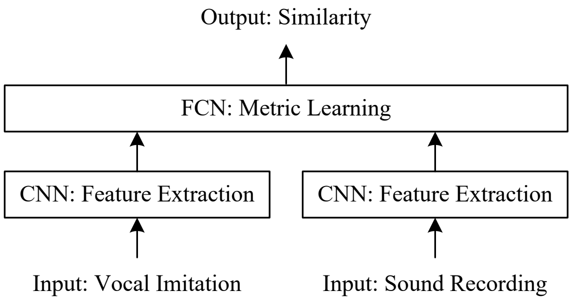


Fig. 1: Basic framework of the proposed Siamese style neural network, SS-CNN, for sound retrieval by vocal imitation. The two CNN feature extraction blocks can be of the same structure (IMINET) or designed and pre-trained differently for the respective inputs (TL-IMINET).

Although the Siamese-style model is trained in a supervised fashion on positive and negative query-candidate pairs, the sound retrieval process is unsupervised, similar to systems that use hand-crafted similarity measures [9], [10], [2]. In other words, the system can be applied to sound candidates and queries that did not appear in the training set, as shown in our experiments in Section VI. This lets the user add new classes of audio to the database and search for them without having to retrain the model.

We will now discuss two architectures and training approaches to building a network to measure similarity between a vocal imitation and an audio file: A symmetric model and an asymmetric one that applies transfer learning. The symmetric model, or IMINET, has two convolutional towers of the same structure. We explore different variants of weight sharing configurations between the two towers in this work, i.e. tied, partially tied, and untied weights. The transfer learning model, or TL-IMINET, has two convolutional towers that are domain-specific. Vocal imitations are generated by human vocal organs, which are closely related to speech audio, so the vocal imitation tower structure originates from a spoken language recognition model [35]. General sound recordings are more varied, hence we design the recording tower structure based on an environmental sound classification model [36]. Rigorously speaking, TL-IMINET is not a Siamese network as its two towers are not identical. However, it is derived from a Siamese network, IMINET, which itself contains three versions with different levels of symmetry. By calling TL-IMINET "Siamese style", we would like to highlight the evolving trend from the most symmetric version, IMINET with tied weights, to IMINET with partially tied and untied weights, and to the least symmetric version, TL-IMINET. This offers a refreshing perspective for the evolution of the network

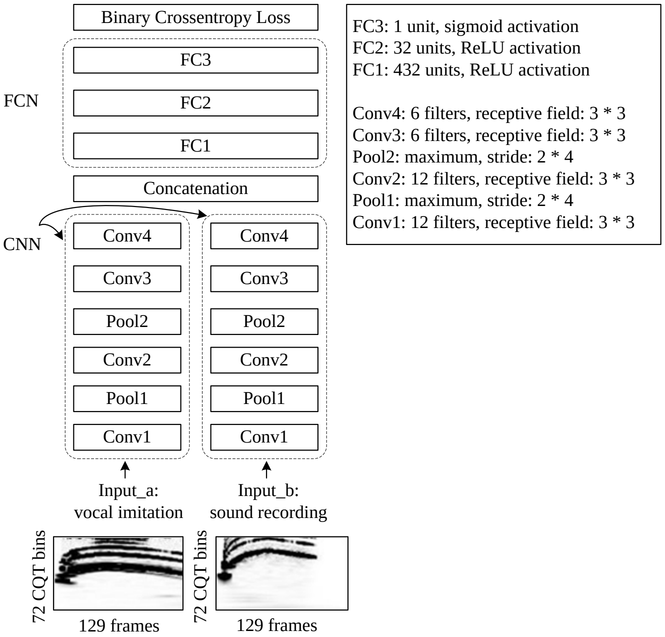

Binary Crossentropy Loss
FC3
FC2
FC1
FCN
Concatenation
CNN
Conv4
Conv3
Pool2
Conv2
Pool1
Conv1
Conv4
Conv3
Pool2
Conv2
Pool1
Conv1
Input_a:
vocal imitation
Input_b:
sound recording
72 CQT bins
129 frames
72 CQT bins
129 frames
FC3: 1 unit, sigmoid activation
FC2: 32 units, ReLU activation
FC1: 432 units, ReLU activation
Conv4: 6 filters, receptive field: 3 * 3
Conv3: 6 filters, receptive field: 3 * 3
Pool2: maximum, stride: 2 * 4
Conv2: 12 filters, receptive field: 3 * 3
Pool1: maximum, stride: 2 * 4
Conv1: 12 filters, receptive field: 3 * 3

Fig. 2: Architecture of the symmetric model, IMINET. The two input spectrograms are of the same size. The two CNN feature extraction towers are of the same structure, yet their weights can be tied, partially tied, or untied.

structure in our research through a wide spectrum of Siamese style networks. Thereform, "Siamese style" in this paper does not necessarily indicate a strict symmetry between the two towers.

IV. Symmetric Model

The symmetric model IMINET is a Convolutional Semi-Siamese Network (CSN). The overall structure is shown in Figure 2. The two towers of convolutional layers for feature extraction share the same structure, yet their weights can be tied, partially tied, or non-tied, making them not fully Siamese.

A. Preprocessing

Both the vocal imitations and original recordings are first downsampled to 16 kHz to have a direct comparison with our previously proposed IMISOUND system [9]. A 6-octave (50-3200 Hz) Constant-Q Transform (CQT) is then employed to calculate their spectrograms using the MATLAB CQT toolbox [37]. The CQT uses 12 bins in each octave and a hop size of 26.25 ms. Considering the fixed size input for convolution in the two towers, both imitation and recording CQT spectrograms truncate the end to maintain 129 frames. Spectrograms shorter than 129 frames are zero-padded. Therefore, the CQT spectrograms have a dimensionality of 72×129 (frequency bins $\times$ time frames). The reasons for using a CQT instead of linear-frequency spectrograms are twofold: 1) the log-frequency scale in a CQT better corresponds to human auditory perception; 2) the representation is more compact compared with linear frequency spectrograms such as STFT for the ease of network training.

B. Feature Learning

Each tower of the Siamese network is a Convolutional Neural Network (CNN) with 4 convolutional layers. The parameters are shown on the upper right side in Figure 2. Both towers receive a 72×129 sized CQT spectrogram as input. Both Conv1 and Conv2 have 12 filters with a receptive field of 3×3 and a stride size of 1×1, followed by a Rectified Linear Unit (ReLU) activation function. They are then each followed by a 2×4 (both shape and stride) max-pooling layer with 2 in frequency and 4 in time, where every time-frequency point in the feature map is covered by exactly one max-pool. For Conv3 and Conv4, each has 6 filters with a receptive field of 3×3 and a stride size of 1×1 followed by ReLU activations, but no pooling layer follows.

Besides sharing the same architecture, Siamese networks usually tie the parameters of the two towers, i.e., the two inputs pass through exactly the same networks for feature learning. This is suitable when the two inputs share many traits, i.e., image matching [28]. In our work, however, vocal imitations lie in a much more restricted sound space than general sound recordings, due to the physical constraints of the human vocal system. Conceptually, vocal imitations and original recordings should pass through two different feature learning networks. Therefore, in IMINET we explore three configurations when designing the convolutional towers:

1) Tied Configuration: The two towers share exactly the same weights and biases in all layers.

2) Untied Configuration: The two towers do not share weights and biases at all, although their structures are the same. This allows the two towers to be tuned for their input domains independently.

3) Partially Tied Configuration: The weights and biases in the two towers are not shared for Conv1 and Conv2 layers, but are shared for Conv3 and Conv4 layers. The rationale behind this design is that layers close to the input should be tuned to adapt to the input's unique characteristics and extract surface-level features that are closely related to the specific input domain, while deeper layers should behave like "grandmother cells" [38] that extract more complex and highly conceptual features [39] that are shared across input domains.

In both untied and partially tied configurations, the symmetry between the two towers are less strict, and we call such structures semi-Siamese networks.

C. Metric Learning

After the features from the two towers are extracted, they are vectorized and concatenated. Then they are fed into a 3-layer Fully Connected Network (FCN), where each unit in layer l is connected to every unit in layer $l-1$. There are 432 neurons in the first layer of the fully connected network (FC1) and 32 in the second layer (FC2). The ReLU activation function is used in both layers. The number of FCN layers and the number of neurons in FC1 and FC2 are chosen after trial and error to achieve the highest retrieval performance on the validation set (see Experiments section). To avoid overfitting, we use 20% dropout on both FC1 and FC2. The third layer (FC3) has only one neuron which uses the sigmoid activation function to squash the output value between 0 and 1. This value is viewed as the similarity between the query-candidate pair.

D. Training

Training the network requires positive and negative pairs of vocal imitations and sound recordings.

We combine vocal imitations with the original sound recordings that they imitate as positive pairs, and with other sound recordings as negative pairs. There exist a total of 840 positive pairs and 840 negative pairs in the training set, without data augmentation. Details about the dataset segmentation are discussed in Section VI.

In our training the ground truth label is 1 for positive pairs and 0 for negative pairs. The loss function we minimize is the binary cross-entropy between the probability output of the network and the binary ground-truth label. We use the Adaptive Moment Estimation (Adam) optimization algorithm [40]. The learning rate is 0.001; β_1 and β_2 are 0.9 and 0.999, respectively; ϵ is 1e-8. The batch size is 128. Early stopping based on validation loss with patience of 5 epochs is employed for training termination. Parameters are chosen by extensive experimentation and fine-tuning for better validation set performance.

Back-propagation is carried out from the FCN to the two Siamese towers. Compared to common distance/similarity measures such as Euclidean distance or cosine similarity, this similarity is learned together with the feature representations of the vocal imitations and original recordings, likely leading to a better retrieval performance.

V. Transfer Learning Model

The partially tied and untied configurations of IMINET introduced flexibilities to the feature extraction towers for them to adapt to their respective inputs, their structures, however, are still the same. In this section, we extend the idea to allow structural differences between the two CNN towers. We introduce the transfer learning idea to pre-train the two CNN towers on their own relevant external tasks, leading to the TL-IMINET model.

The overall structure of TL-IMINET is shown in Figure 3. It is also a Siamese style convolutional neural network, but the structure is not as symmetric as IMINET. The recording and imitation towers for feature extraction are adapted from environmental sound classification and spoken language recognition tasks, respectively, hence are asymmetric. The two tower weights and biases are initialized by pre-training them on external datasets for these tasks. They are then fine-tuned together with the FC layers on the sound retrieval task. At test time, the sound retrieval procedure is the same as IMINET: The network output is a similarity value between an imitation query and an original sound candidate from the search database. Sound candidates with highest similarity values are returned.

As a new task, sound retrieval through vocal imitation suffers from the data scarcity issue, therefore, we hope to use transfer learning to alleviate the problem. Transfer learning has the benefit of passing knowledge learned from related relevant tasks that are often data-rich to the task at hand, which is often data-hungry. The key consideration is to find appropriate relevant data-rich tasks to transfer knowledge from. In the following subsections, we will describe our design in detail.

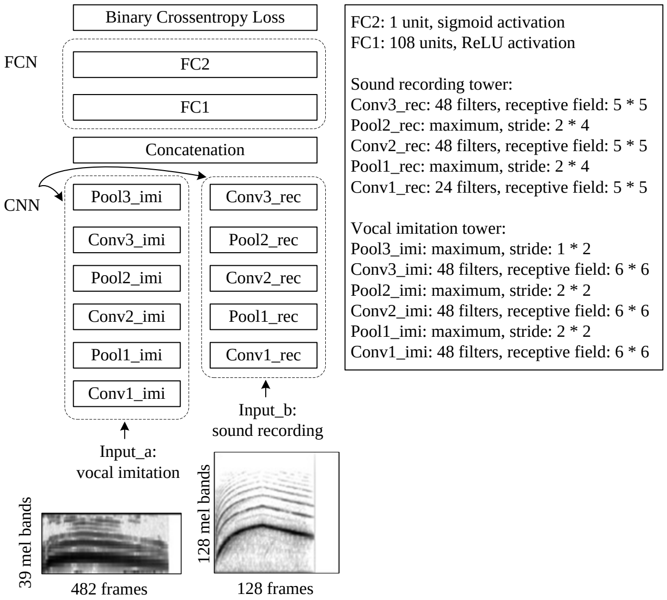

Binary Crossentropy Loss
FCN
FC2
FC1
Concatenation
CNN
Pool3_imi
Conv3_imi
Pool2_imi
Conv2_imi
Pool1_imi
Conv1_imi
Conv3_rec
Pool2_rec
Conv2_rec
Pool1_rec
Conv1_rec
Input_a:
vocal imitation
Input_b:
sound recording
39 mel bands
482 frames
128 mel bands
128 frames
FC2: 1 unit, sigmoid activation
FC1: 108 units, ReLU activation
Sound recording tower:
Conv3_rec: 48 filters, receptive field: 5 * 5
Pool2_rec: maximum, stride: 2 * 4
Conv2_rec: 48 filters, receptive field: 5 * 5
Pool1_rec: maximum, stride: 2 * 4
Conv1_rec: 24 filters, receptive field: 5 * 5
Vocal imitation tower:
Pool3_imi: maximum, stride: 1 * 2
Conv3_imi: 48 filters, receptive field: 6 * 6
Pool2_imi: maximum, stride: 2 * 2
Conv2_imi: 48 filters, receptive field: 6 * 6
Pool1_imi: maximum, stride: 2 * 2
Conv1_imi: 48 filters, receptive field: 6 * 6

Fig. 3: Architecture of the asymmetric model, TL-IMINET. The two input spectrograms are of different sizes, and pass through different neural network structures in the two CNN towers for feature extraction.

A. Imitation Tower Pre-training

Vocal imitations are produced by the human vocal system and share some similar acoustical characteristics with speech utterances. Here we use a spoken language classification task to pre-train the vocal feature extraction tower of TL-IMINET. Compared to other speech processing tasks such as speech recognition and sining classification, the audio materials in language classification contain various kinds of phonemes among different languages and are much richer. This richness is preferred as vocal imitations are freely generated using a variety of vocal organs such as tongues, cheeks, and teeth.

We adopt the CNN architecture proposed in [35] with slight modifications. The original system segments the audio signal into 5-second long windows, and feeds each window to the network. It encodes the audio into a 39-band log-mel spectrogram with a frame hop size of 8.33 ms, where filter center frequencies range between 0 and 5 kHz. It then uses a 3-layer CNN followed by 2 FC layers to classify the input into three classes: English, French, and German.

Our modified structure is shown in Figure 4. First, we reduce the speech signal window size to 4 seconds because most vocal imitations in the data set are less than 4 seconds long. Then each 4 second speech is converted to a 39-band log-mel spectrogram with an 8.33 ms non-overlapping analysis window in accordance with [35]. Therefore, the final spectrogram has a dimensionality of 39 frequency bins in by

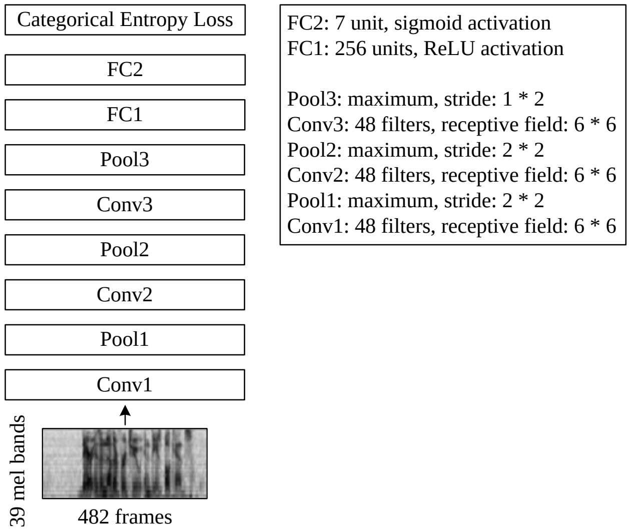

Categorical Entropy Loss
FC2
FC1
Pool3
Conv3
Pool2
Conv2
Pool1
Conv1
39 mel bands
482 frames
FC2: 7 unit, sigmoid activation
FC1: 256 units, ReLU activation
Pool3: maximum, stride: 1 * 2
Conv3: 48 filters, receptive field: 6 * 6
Pool2: maximum, stride: 2 * 2
Conv2: 48 filters, receptive field: 6 * 6
Pool1: maximum, stride: 2 * 2
Conv1: 48 filters, receptive field: 6 * 6

Fig. 4: The neural network structure derived from [35] with slight modifications. The input is a log-mel spectrogram with 39 mel frequency bands (0-5 kHz) and 482 time frames (4 seconds). The exemplar spectrogram represents a male speech in English.

482 time steps (39×482). For layers Conv1 and Conv2, each layer has 48 filters with ReLU activations, and each is followed by a 6×6 pooling layer. Conv3 also has 48 filters and a receptive field of 6×6, followed by a 1×2 pooling layer with 1 in frequency and 2 in time. More detailed parameters are described in the figure.

We pre-train this network on VoxForge, a free speech corpus and acoustic model repository for open source speech recognition engines [41], [35]. It contains user-uploaded speech recordings in different languages, in both 8 kHz and 16 kHz sampling rates. We use 16 kHz recordings in this work. We choose seven languages to construct a 7-class classification task: Dutch, English, French, German, Italian, Russian, and Spanish, which have the most recordings. For each language, we choose 8,000 speech clips (about 4 seconds long on average) from different people. This dataset is split into 70% for training and 30% for testing. After training, our model achieves 69.8% accuracy on the test set, which is relatively good for a 7-class classification task, compared with the reported 80.1% accuracy for a 3-class classification task in [35].

B. Recording Tower Pre-training

The original sound recordings in our dataset represent a large number of concepts, generated by various sound sources. The design of our tower for the original recordings is based on a CNN architecture used for environmental sound classification [36]. In this task, an audio clip to be classified (3 seconds long) is first converted into a log-mel spectrogram with a 23 ms non-overlapping analysis window, with frequency range of 0 to 22,050 Hz. This leads to a input dimensionality of 128×128, representing 128 mel-frequency bands and 128 frames in time.

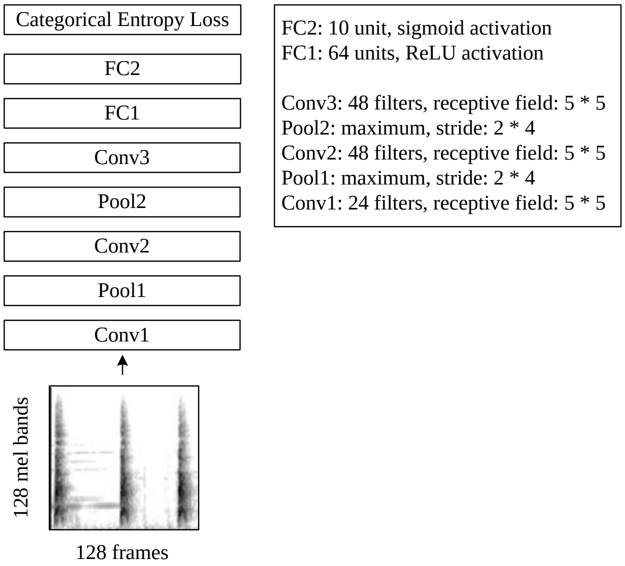

Categorical Entropy Loss
FC2
FC1
Conv3
Pool2
Conv2
Pool1
Conv1
128 mel bands
128 frames
FC2: 10 unit, sigmoid activation
FC1: 64 units, ReLU activation
Conv3: 48 filters, receptive field: 5 * 5
Pool2: maximum, stride: 2 * 4
Conv2: 48 filters, receptive field: 5 * 5
Pool1: maximum, stride: 2 * 4
Conv1: 24 filters, receptive field: 5 * 5

Fig. 5: The neural network structure in [36]. The input is a log-mel spectrogram with 128 mel frequency bands and 128 frames. The exemplar spectrogram represents a "dog bark" sound event for 3 seconds.

The spectrogram is fed into a convolutional neural network with 3 convolutional layers and 2 fully connected layers. The neural network structure is shown in Figure 5. Conv1 has 24 filters with a receptive field of 5×5, and followed by a ReLU activation function. They are then followed by a 2×4 (both shape and stride) max-pooling layer with 2 in frequency and 4 in time. For Conv2 and Conv3, each has 48 filters with a 5×5 receptive field. Conv2 is followed by a 2×4 pooling layer Pool2, but no pooling layer follows Conv3. Then the activations of Conv3 are followed by a 64-neuron fully connected layer FC1 and the final output layer FC2 has 10 neurons, indicating probabilities of predicting to one of ten classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music.

We replicated the experiments in [36], by a 10-fold cross validation without data augmentation. The system achieves 70.2% accuracy on average, which is close to the accuracy reported in [36]. This gives us confidence that the network likely learned structures useful for general audio classification.

C. Metric Learning

In both pre-training tasks the fully-connected layers are removed while the convolutional layer weights are applied as an initialization for the two towers of TL-IMINET, which serve as feature extractors for imitation and original sounds. After the features from the two towers are extracted, they are flattened and concatenated. Then they are fed into a 2-layer Fully Connected Network (FCN). There are 108 neurons in FC1 and the ReLU activation function is adopted to avoid vanishing gradient. The number of FCN layers and number of FC1 neurons are chosen after trial and error to achieve the highest retrieving performance (see Experiments section) on the validation set. In FC2 there is only 1 neuron with the sigmoid activation function to squash the output value between 0 and 1. Similar to the symmetric model, this value is viewed as the similarity between the imitation-recording input pair.

D. Training

Training for TL-IMINET is similar to that for IMINET described in Section IV-D: We generate positive and negative imitation-original pairs and minimize the cross-entropy loss between the probability prediction and the ground-truth labels. The main difference is that for IMINET, all network weights/biases are randomly initialized, while for TL-IMINET, some CNN layers of the feature extraction towers are initialized with the pre-trained weights in Sections V-A and V-B.

By varying the number of pre-trained layers, we can investigate the effect of pre-training on the sound retrieval task. As there are 3 convolutional layers in both the recording and imitation towers, and considering that early layers are more appropriate to be pre-trained on other tasks, we apply three different pre-training configurations for TL-IMINET:

1) No pre-training: All network weights are randomly initialized. Transfer learning is not applied.

2) Pre-train Conv1: Only Conv1 weights of both towers are initialized with pre-trained weights.

3) Pre-train Conv1/2: Only Conv1 and Conv2 of both towers are initialized with pre-trained weights.

4) Pre-train Conv1/2/3: All three Conv layers of both towers are initialized with pre-trained weights.

The other difference from IMINET is that, we employ Stochastic Gradient Descent (SGD) optimization algorithm to minimize the loss function of binary cross-entropy between the probability (similarity) output and the ground-truth label, where 1 and 0 denotes positive and negative pairs, respectively. For TL-IMINET, we observe that SGD achieves better sound retrieval performance compared with Adam. The learning rate is 0.01, learning rate decay is 0.0001, and momentum is 0.9. The batch size is 128 and training is terminated after 30 epochs. The above hyper parameters as well as hidden layer sizes, kernel sizes and pooling sizes are chosen through extensive experimentation to achieve high performance on the validation set. This is not an exhaustive grid search of the parameter combinations, but rather a search among a number of randomly selected combinations.

VI. Experiments

In this section, we would like to answer the following questions through experiments and analyses: 1) How do the various versions of the proposed SS-CNN model compare with the state-of-the-art baseline, IMISOUND? 2) Does transfer learning from other relevant tasks improve SS-CNN's sound retrieval performance? 3) Can we further improve the performance by fusing different configurations of SS-CNN and perhaps with IMISOUND as well?

A. Dataset

We use the VocalSketch Data Set v1.0.4 [8] in our experiments. This dataset contains hundreds of original sounds, each representing a distinct concept, and 10 vocal imitations of each sound obtained from different Amazon Mechanical Turk users. The sounds and imitations are 3-second long on average. The sounds fall into 4 broad categories, namely Acoustic Instruments (AI), Commercial Synthesizers (CS), Everyday (ED), and Single Synthesizer notes (SS). The number of sounds in these categories is 40, 40, 120, and 40, respectively. We choose half of the sounds of each category (i.e., 20, 20, 60, and 20 from AI, CS, ED, SS, respectively) and all of their imitations to compose a dataset to train and validate our models. We use the other half of sounds and their imitations to test the models. Therefore, training and testing materials do not share any sounds nor imitations.

For the 120 sounds used for training and validation, we choose 7 imitations of each sound to form $120 \times 7 = 840$ positive pairs and 840 negative pairs to train both IMINET and TL-IMINET. Positive pairs are pairs of an imitation and its target sound. Negative pairs are created by randomly pairing an imitation with an irrelevant sound. We use the remaining 3 imitations of each sound to compose $120 \times 3 = 360$ positive pairs and 360 negative pairs to validate the IMINET and TL-IMINET. The total amount of training and validation pairs are of 3.3 and 1.4 hours in time. We then evaluate the sound retrieval performance (see Section VI-B) of different methods within each category of the remaining 120 sounds and their imitations, taking each imitation as the query and averaging the retrieval performance.

B. Evaluation Measures

We employ Mean Reciprocal Rank (MRR) [42] to evaluate the search performance in each category:

$$MRR = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_i}, \tag{1}$$

where $rank_i$ is the rank of the target sound among all sounds in the same category for the i-th vocal imitation query; Q is the number of imitations in each category. MRR ranges from 0 to 1 with a higher value indicating a better sound retrieval performance. For example, an MRR of 0.5 suggests that, on average, the target sound is ranked the second among all sounds in the category. For each method, we report the average MRR and standard deviation across 10 models trained with different initializations. We compare with our previous IMISOUND system [10], which achieved the state-of-the-art MRR performance for sound retrieval through vocal imitation on the VocalSketch dataset.

C. Baseline Method

We choose the previous state-of-the-art system for sound retrieval by vocal imitation, IMISOUND [9], [10], as the baseline for comparison. In this system, vocal imitations and sound recordings are processed in the same way as the IMINET that they are first downsampled to 16 kHz and then converted to 6-octave (50-3200 Hz) CQT spectrograms using [37]. The spectrogram is segmented into overlapping 525 ms long patches. Then a two-hidden-layer Stacked Auto-Encoder (SAE) [43] is employed as a feature extractor applied to the vectorized patches. The first and second hidden layer have 1,000 and 600 neurons, respectively. Each patch is then represented as a 600-d vector. We further include its first-order derivative (delta) w.r.t. time, resulting in a 1,200-d vector for each patch. Each vocal imitation and sound candidate is thus represented by a sequence of 1,200-d vectors. In order to get the recording-level feature representation, we calculate maximum, minimum, mean, median, standard deviation, and interquartile range within each dimension. Finally, each vocal imitation and sound candidate is represented by a 7,200-d vector. Cosine similarities between the vocal imitation query and all sound candidates within a category are calculated using the feature representation.

D. Fusion Strategies

Inspired by ensemble learning [44], we consider the idea of fusing the retrieval results of the different configurations of SS-CNN by multiplying their similarity outputs (i.e., probability of being a positive pair). This is similar to what naive Bayes does on fusing predictions made along different dimensions. Fusion was first applied in IMINET [5] and it is now also used for TL-IMINET.

1) Fusion for IMINET: We fuse the three different configurations of tied, partially tied, and untied weights of IMINET. Specifically we have:

$$L_{fusion1}(i) = L_{tied}(i) * L_{untied}(i) * L_{partial}(i), \quad (2)$$

where $L_{tied}(i)$, $L_{untied}(i)$, and $L_{partial}(i)$ are the pairing likelihood between the query and the i-th sound candidate, by tied, untied, and partially tied models, respectively.

We also consider to fuse the retrieval results of IMINET with those of IMISOUND [10]. As described before, IMISOUND uses a two-hidden-layer SAE to extract features for a vocal imitation and a sound candidate. It then calculates the cosine distance between their feature representations. To fuse this result with that of IMINET, we convert the distance to a likelihood through a softmax function:

$$L_{sae}(i) = \frac{e^{-D(i)}}{\sum_{n=1}^{N} e^{-D(n)}}, \quad (3)$$

where $D(i)$ is the cosine distance between the vocal imitation and the i-th sound candidate; N is the total number of sound candidates in the library. Then the fusion between IMINET and IMISOUND can be done by multiplying their likelihood values:

$$L_{fusion2}(i) = L_{iminet}(i) * L_{sae}(i). \quad (4)$$

2) Fusion for TL-IMINET: We fuse each pre-training configuration with IMISOUND, similar as Equation (4):

$$L_{fusion3}(i) = L_{tl-iminet}(i) * L_{sae}(i). \quad (5)$$

Siamese style networks from our current work (IMINET and TL-IMINET) and previous work (IMISOUND) have different structures and training objectives. In particular, the SS-CNN networks feature representations in a supervised way with the goal of helping discriminate positive and negative pairs, while IMISOUND learns features in an unsupervised way which aims at a good reconstruction of the input. In addition, Siamese style networks learn the similarity between vocal imitations and sound recordings from training data, while IMISOUND uses a pre-defined distance measure. Therefore, it is expected that they perform differently on the same imitation-sound pair and fusing their results may improve the retrieval performance.

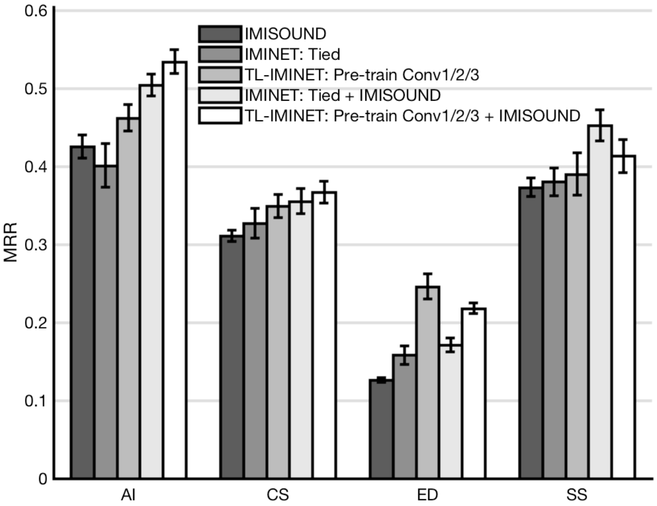


Fig. 6: Sound retrieval performance comparisons among IMISOUND, IMINET with tied weights, TL-IMINET with pre-training Conv1/2/3, IMINET with tied weights fused with IMISOUND, and TL-IMINET with pre-training Conv1/2/3 fused with IMISOUND.

E. Experimental Results

Table 1 shows comprehensive performance comparisons of the IMISOUND baseline [10], different configurations of the proposed IMINET, TL-IMINET, and their different fusing strategies in five groups (blocks). Considering the large amount of configurations of the proposed methods, we will first analyze the configurations of IMINET and TL-IMINET separately. We will then choose the best configurations of IMINET and TL-IMINET and compare them to demonstrate the advantages of transfer learning. Finally, we will analyze the benefits of fusing IMINET/TL-IMINET and IMISOUND.

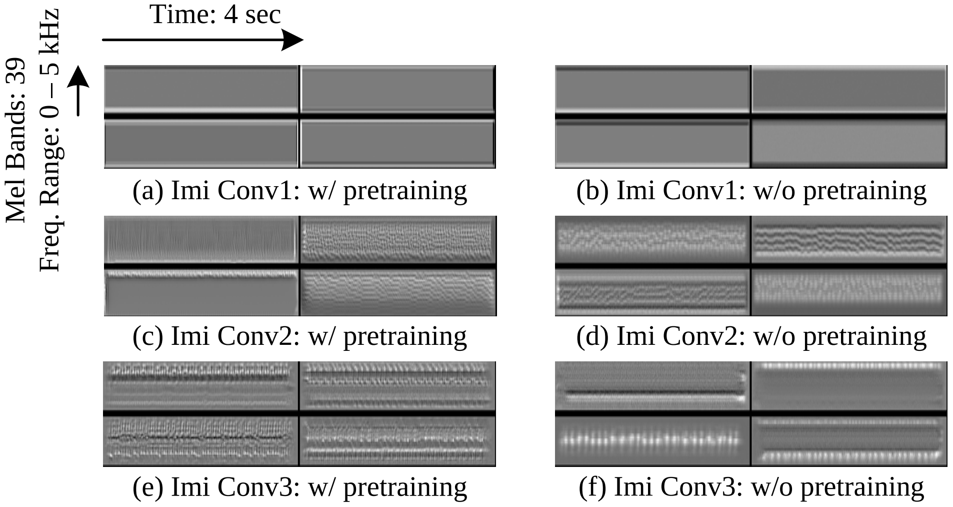
1) Configuration Comparison for IMINET: We compare the three weight sharing strategies (untied, partially tied, and tied) for IMINET and the fusion of the three systems. Several interesting observations can be made from Table I.

First, from untied to partially tied to tied configurations of IMINET, the MRR increasing trend is observed in all categories. This is unexpected, as we thought that partially tied or untied configuration could better account for the differences between vocal imitations and sound recordings and result in better retrieval performance. A possible explanation could be that the number of parameters is reduced in the tied configuration, which makes the network easier to train considering the small amount of training data. This suggests that data scarcity might be a bottleneck hindering the potential exploitation of more complicated models.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2868428, IEEE/ACM Transactions on Audio, Speech, and Language Processing

TABLE I: MRR (mean ± std) comparisons of the baseline system (IMISOUND), various configurations of the proposed IMINET and TL-IMINET, and different fusion systems of IMINET and IMISOUND. Higher values are better and the best results for each category in each block are in bold. Some results of IMINET are from [5].

	Configuration	Acoustic Instr.	Comm. Synthesizers	Everyday	Single Synthesizer
Baseline	IMISOUND	0.425 ± 0.015	0.311 ± 0.007	0.126 ± 0.003	0.373 ± 0.012
IMINET	Untied	0.377 ± 0.019	0.318 ± 0.020	0.154 ± 0.014	0.325 ± 0.020
	Partial	0.384 ± 0.027	0.304 ± 0.015	0.154 ± 0.015	0.340 ± 0.031
	Tied	0.401 ± 0.028	0.327 ± 0.019	0.158 ± 0.012	0.380 ± 0.018
	Untied + Partial + Tied	**0.438 ± 0.015**	**0.343 ± 0.020**	**0.175 ± 0.012**	**0.382 ± 0.013**
TL-IMINET	No Pre-train	0.397 ± 0.027	0.309 ± 0.021	0.225 ± 0.023	0.377 ± 0.025
	Pre-train Conv1	0.412 ± 0.033	0.328 ± 0.027	0.227 ± 0.020	0.399 ± 0.036
	Pre-train Conv1/2	0.432 ± 0.024	0.325 ± 0.023	0.225 ± 0.016	**0.404 ± 0.036**
	Pre-train Conv1/2/3	**0.462 ± 0.017**	**0.349 ± 0.015**	**0.246 ± 0.016**	0.390 ± 0.027
IMINET + IMISOUND	Untied + IMISOUND	0.470 ± 0.025	0.356 ± 0.011	0.168 ± 0.010	0.402 ± 0.022
	Partial + IMISOUND	0.496 ± 0.018	0.346 ± 0.025	0.173 ± 0.014	0.417 ± 0.025
	Tied + IMISOUND	0.504 ± 0.014	0.355 ± 0.016	0.171 ± 0.009	**0.452 ± 0.020**
	Untied + Partial + Tied + IMISOUND	**0.520 ± 0.020**	**0.371 ± 0.013**	**0.188 ± 0.007**	0.447 ± 0.012
TL-IMINET + IMISOUND	No Pre-train + IMISOUND	0.490 ± 0.017	0.339 ± 0.017	0.199 ± 0.013	0.429 ± 0.025
	Pre-train Conv1 + IMISOUND	0.513 ± 0.029	0.352 ± 0.026	0.198 ± 0.014	**0.441 ± 0.023**
	Pre-train Conv1/2 + IMISOUND	0.519 ± 0.014	0.353 ± 0.016	0.209 ± 0.008	0.429 ± 0.028
	Pre-train Conv1/2/3 + IMISOUND	**0.534 ± 0.015**	**0.367 ± 0.014**	**0.218 ± 0.007**	0.413 ± 0.021



(a) Imi Conv1: w/ pretraining

(b) Imi Conv1: w/o pretraining

(c) Imi Conv2: w/ pretraining

(d) Imi Conv2: w/o pretraining

(e) Imi Conv3: w/ pretraining

(f) Imi Conv3: w/o pretraining

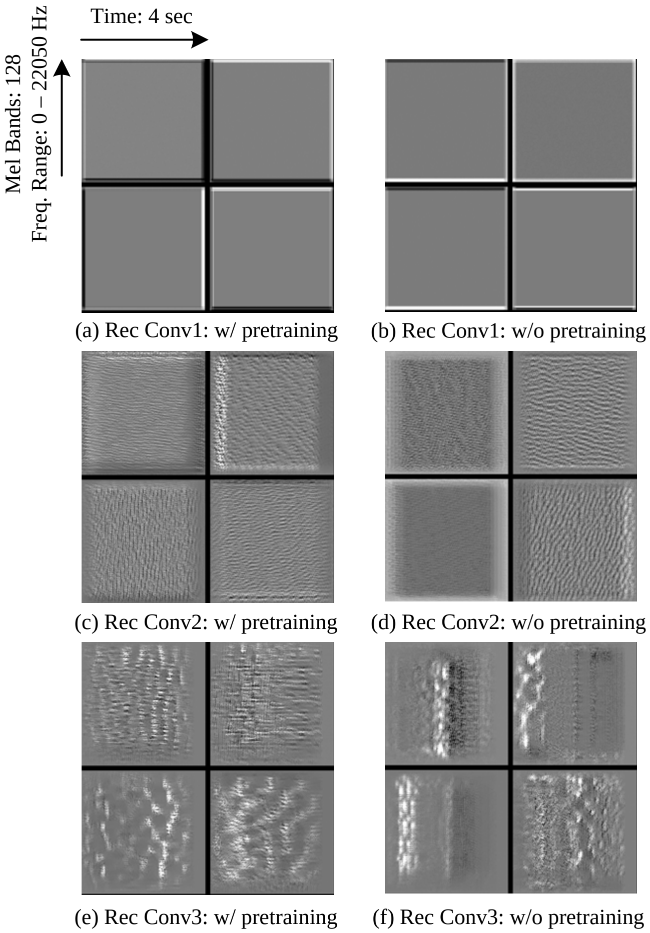
Fig. 7: Visualization of input patterns that maximally activate four randomly selected neurons from each of the three convolutional layers (Conv1, Conv2, and Conv3) in the imitation tower of TL-IMINET, without pre-training (left column) and with pre-training using the Vox Forge data set (right column). Whiter color indicates higher energy.

Second, the best performing IMINET configuration, tied, outperforms the IMISOUND baseline on two categories (Commercial Synthesizers and Everyday), underperforms on the Acoustic Instrument category, and achieves comparable performance on the Single Synthesizer category. Unpaired t-tests show that the MRR improvement is statistically significant for Commercial Synthesizers (p = 1.17e-2) and Everyday (p = 6.15e-6) at the significance level of 0.05.

Third, by fusing different configurations of IMINET, the MRR is better than each configuration itself. The MRR improvements for all categories except Signal Synthesizer are statistically significant (Acoustic Instruments p = 3.08e-2, Commercial Synthesizers p = 2.70e-4, and Everyday p = 8.30e-8), at the significance level of 0.05, under unpaired t-tests. This is because under different weight constraints, each configuration tends to learn its unique features. We believe that these features are complementary to some extent, explaining why the fused model outperforms every single configuration.

2) Configuration Comparison for TL-IMINET: For the proposed TL-IMINET, we compare its different pre-training strategies with the IMISOUND baseline.

First, we see that TL-IMINET without pre-training outperforms IMINET untied in all categories except the Commercial Synthesizers category. It is noted that the main difference between these two models is on the network structure of the convolutional towers: IMINET uses the same structure for both the imitation and original sound towers, while TL-IMINET uses different structures that are originally designed for the spoken language classification and environmental sound classification tasks, respectively. This suggests that using structures that are carefully designed for the different types of sounds (voices vs. general sounds) achieves better sound retrieval



(a) Rec Conv1: w/ pretraining (b) Rec Conv1: w/o pretraining

(c) Rec Conv2: w/ pretraining (d) Rec Conv2: w/o pretraining

(e) Rec Conv3: w/ pretraining (f) Rec Conv3: w/o pretraining

Fig. 8: Visualization of input patterns that maximally activate four randomly selected neurons from each of the three convolutional layers (Conv1, Conv2, and Conv3) in the recording tower of TL-IMINET, without pre-training (left column) and with pre-training using UrbanSound8k data set (right column). Whiter color indicates higher energy.

performance than using non-informative structures that do not consider the differences between the types of sounds. The reason that the Everyday category receives the most prominent improvement may be due to the fact that the original tower structure is used to train environmental sounds [36], which is expected to work well in Everyday category that share many characteristics with the UrbanSound8K dataset.

Second, comparing different pre-training strategies of TL-IMINET, we see a relatively clear trend of MRR increase from no pre-training to pre-training more layers. We observe that in the third block, the mean MRR increases from no pre-training to pre-training more convolutional layers, and pre-training all convolutional layers achieves the highest MRR scores in all categories except Single Synthesizer. This finding supports our assumption that transfer learning from relevant tasks is helpful to our sound retrieval task.

Third, the best performing TL-IMINET configuration, pre-training all 3 convolutional layers on both towers, significantly outperforms the baseline IMISOUND in all categories, Acoustic Instruments (p = 2.57e-9), Commercial Synthesizers (p = 7.49e-4), Everyday (p = 5.21e-11), and Single Synthesizer (p = 8.35e-6), according to a set of unpaired t-tests at the significance level of 0.01. Even the no pre-training TL-IMINET outperforms IMISOUND at the significance level of 0.05, for Acoustic Instruments (p = 3.65e-2), Everyday (p = 3.57e-7), and Single Synthesizer (p = 4.81e-5).

3) Fusing IMINET with IMISOUND: As described in Section IV, the proposed IMINET framework has a very different design from the baseline IMISOUND system: IMISOUND uses unsupervised learning (stacked auto-encoders) to learn feature representations from training imitations and then uses pre-defined similarity measures to match imitations and original sounds, while IMINET learns feature representations and similarity measures simultaneously from positive and negative training pairs in a supervised fashion. It is thus possible that IMISOUND and IMINET behave complimentarily and fusing them may improve the performance.

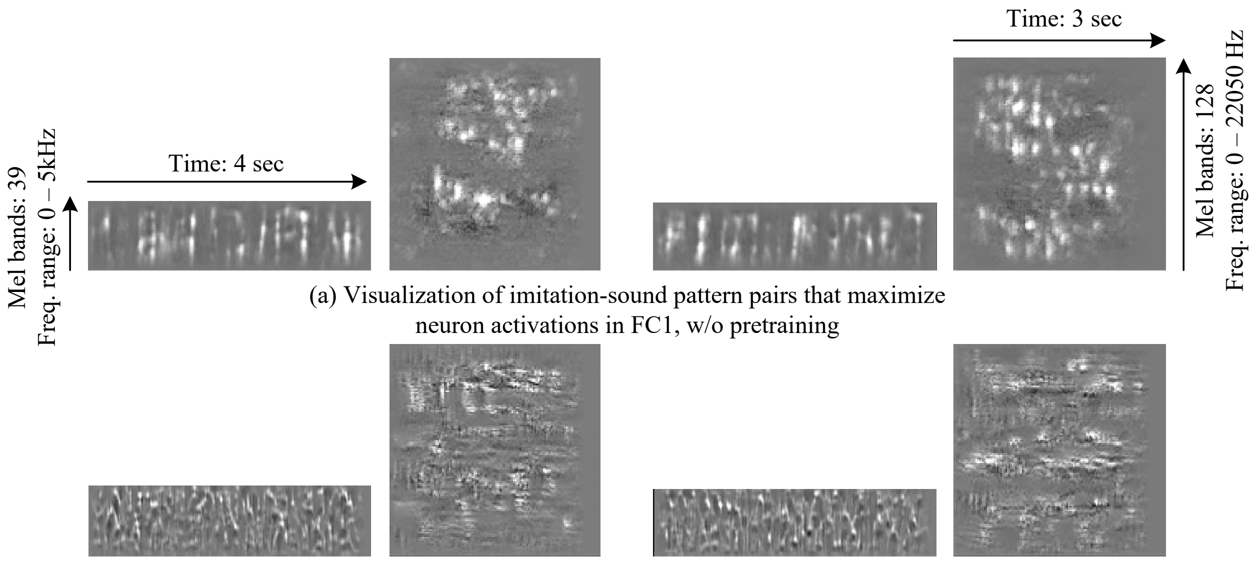
This hypothesis is validated by comparing the second and third blocks against the fourth and fifth blocks of Table I. All configurations of IMINET show a significant improvement of MRR after they are fused with IMISOUND. Similarly, all configurations of TL-IMINET except Everyday category also show a significant improvement of MRR after they are fused with IMISOUND. The above improvements are all statistically significant under a set of unpaired t-tests at the significance level of 0.05.

To make this observation clearer, we choose the best configuration of IMINET (tied) and TL-IMINET (Pre-train Conv1/2/3), respectively, and compare them with IMISOUND as well as their fusion with IMISOUND. This comparison is shown in Figure 6. We can observe that, 1) there is a clear trend that Siamese style neural networks outperform IMISOUND, 2) Fusion with IMISOUND helps to improve both IMINET and TL-IMINET performance in almost every category but TL-IMINET in Everyday category, and 3) overall, TL-IMINET fused with IMISOUND works the best across all categories.

VII. Visualization and Sonification

In order to obtain more insights on how SS-CNN works, in this section we visualize and sonify the input patterns that maximize the activation of certain neurons in each layer, using the activation maximization approach [32]. We choose TL-IMINET for this analysis.

Activation maximization [32] can be done by gradient ascent of the neuron's activation w.r.t. the input from a random initialization, while keeping the trained weights unchanged. After convergence, the updated input spectrogram can be interpreted as what the neuron learns. For better visualization purposes, ReLU activations in TL-IMINET are replaced by leaky ReLU with a slope of 0.3 for negative inputs. This is to prevent the zero gradient issue when the input value to the ReLU activation is negative, which will trap the optimization. We further sonify the generated input magnitude spectrograms by recovering the phase information using the Griffin-Lim algorithm [45]. The visualization for all input patterns and their corresponding sonified waveforms can be accessed via: https://goo.gl/Y5ytv6.



(a) Visualization of imitation-sound pattern pairs that maximize neuron activations in FC1, w/o pretraining

(b) Visualization of imitation-sound pattern pairs that maximize neuron activations in FC1, w/ pretraining

Fig. 9: Imitation-recording input pair pattern visualization in FC1 of TL-IMINET. Whiter color represents higher energy. Note that they are not positive or negative pairs but input patterns activating a certain FC1 neuron the most.

1) Imitation Tower: The left and right column of Figure 7 shows the filter visualization of the imitation tower with and without pre-training, respectively. First, taking the top left corner pattern in the left column as an example, the horizontal and vertical dimension represents the number of time frames and frequency bins in mel scale, respectively. Whiter color represents higher energy. Note that first layer (Conv1) neurons learn local features like edges, intermediate layer (Conv2) neurons capture more complicated information such as texture with various directions, while the deepest layer (Conv3) neurons recognize spectrogram-like patterns, with a concentration on different frequency ranges. Second, input patterns visualized with pre-training are generally shaper and contain more finer patterns compared with those without pre-training. This suggests that pre-training on the VoxForge dataset helps the feature extraction tower to pay more attention to spectral details. The sonifications in Conv1 sound like low frequency humming, in Conv2 we hear more spectral components, and in Conv3 delicate birding chirping and water flowing like sounds can be heard.

2) Recording Tower: Figure 8 shows input patterns that maximally activate several neurons in the original sound recording tower, with and without pre-training. Interesting findings are also observed: First, we discover the same trend of pattern complexity from shallow layers to deep layers, with input patterns from simple and oriented edges to various texture-like patterns, eventually to spectrogram-like complex and hierarchical structures. Second, dissimilar with what we observed earlier in Conv3 of the imitation tower, Conv3 input patterns of the recording tower tend to learn vertical strips besides horizontal patterns. We note that such input patterns resemble feature maps used by mammals in their auditory systems [46], [47]. When more complex stimuli are provided, early auditory responses progressively show simple-to-structural periodic patterns along time and frequency directions in auditory spectrograms similar to our visualization results in different neural network layers. For the recording tower sonification, in Conv1 we can hear simple sound patterns like constant pitch and spike, in Conv2 we can hear fast changing patterns in time, and in Conv3 modulated sound effects can be heard.

3) Dense Layers: Dense layer filters can be visualized using activation maximization as well. A neuron in a fully connected layer receives a pair of inputs, and the receptive field of each neuron covers the entire input ranges of both the vocal imitation and original recording. Therefore, the neuron is maximally activated by an imitation-original pair instead of an imitation or a original recording alone. This is different from the Single-Input-Single-Output (SISO) network structure where activation maximization was originally applied in [32]. In Figure 9, we show the maximal activation patterns for 2 representative neurons in layer FC1. The corresponding imitation-original input pattern pairs are shown without and with pre-training TL-IMINET respectively. By pre-training TL-IMINET, more detailed structures from the pairs can be observed compared with the configuration of without pre-training. In both Figure 9(a) and (b), imitation and recording show somewhat similar textures to form a pair. By sonifying the imitation-recording input pattern pairs, we hear that the recovered imitation and original sounds are similar from the aspect of temporal evolution but with timbre being different. The recovered imitation sound is more like natural sound (e.g., generated by certain animals) while the recovered recording sound is similar to a robot voice.

VIII. Conclusions

In this paper, we proposed a general Siamese Style Convolutional Neural Network (SS-CNN) model for sound search by vocal imitation. It contains two similar encoders whose structures can be suited to the respective input, which are a vocal imitation query and an original sound from the database to be searched. The two encoders use CNN for feature extraction from input spectrograms, and then learned features are concatenated and fed into a FCN for similarity measure between the two inputs.

By introducing different levels of symmetry, we present two versions of the proposed SS-CNN system: 1) Training-from-scratch IMINET where the two encoders share exactly the same structure, although the encoder weights may be different. 2) Transfer learning based TL-IMINET, where the two encoders use different and domain-specific structures. The encoders are also pre-trained separately on the original tasks before being fine-tuned on the VocalSketch data set.

Experiments show that the proposed Siamese style networks outperform state-of-the-art system IMISOUND where feature learning and matching algorithms are optimized separately. It shows that transfer learning, as well as fusion of the proposed models with IMISOUND significantly improves the system performance. To provide insights to the proposed networks, we visualize and sonify input patterns that maximally excite certain neurons and filters.

For future work, we would like to employ Recurrent Neural Networks (RNN) under the Siamese Network architecture to better model the temporal evolution of both vocal imitations and original sound recordings. To improve the practical usability in large-scale databases, we plan to integrate vocal imitation-based and text-based search together. Finally, we plan to conduct subjective studies on the system usability.

Acknowledgment

This work is funded by the National Science Foundation grants No. 1617107 and No. 1617497. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

[1] Y. Zhang and Z. Duan, "Retrieving sounds by vocal imitation recognition," in *Proc. Machine Learning for Signal Processing (MLSP), 2015 IEEE International Workshop on*, 2015, pp. 1–6.

[2] M. Helén and T. Virtanen, "Audio query by example using similarity measures between probability density functions of features," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, no. 1, p. 179303, 2009.

[3] http://www.cartalk.com [Accessed 05/20/2018].

[4] D. S. Blancas and J. Janer, "Sound retrieval from voice imitation queries in collaborative databases," in *Proc. Audio Engineering Society 53rd International Conference on Semantic Audio*, 2014, pp. 1–6.

[5] Y. Zhang and Z. Duan, "IMINET: Convolutional semi-Siamese networks for sound search by vocal imitation," in *Proc. Applications of Signal Processing to Audio and Acoustics (WASPAA), 2017 IEEE Workshop on*, 2017 (accepted).

[6] ——, "Visualization and interpretation of Siamese style convolutional neural networks for sound search by vocal imitation," in *Proc. Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*, 2018, pp. 2406–2410.

[7] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a 'siamese' time delay neural network," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 1994, pp. 737–744.

[8] M. Cartwright and B. Pardo, "Vocalsketch: Vocally imitating audio concepts," in *Proc. the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 43–46.

[9] Y. Zhang and Z. Duan, "IMISOUND: an unsupervised system for sound query by vocal imitation," in *Proc. Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2016, pp. 2269–2273.

[10] ——, "Supervised and unsupervised sound retrieval by vocal imitation," *Journal of the Audio Engineering Society*, vol. 64, no. 7/8, pp. 533–543, 2016.

[11] M. M. Zloof, "Query-by-example: A data base language," *IBM Systems Journal*, vol. 16, no. 4, pp. 324–343, 1977.

[12] T. Bertin-Mahieux and D. P. Ellis, "Large-scale cover song recognition using hashed chroma landmarks," in *Proc. Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*, 2011, pp. 117–120.

[13] T. K. Chia, K. C. Sim, H. Li, and H. T. Ng, "A lattice-based approach to query-by-example spoken document retrieval," in *Proc. the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 363–370.

[14] A. Wang, "An industrial strength audio search algorithm," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2003, pp. 7–13.

[15] Z. Rafii, B. Coover, and J. Han, "An audio fingerprinting system for live version identification using image processing techniques," in *Proc. Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 644–648.

[16] T. J. Tsai, T. Prätzlich, and M. Müller, "Known artist live song id: A hashprint approach," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 427–433.

[17] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by humming: musical information retrieval in an audio database," in *Proc. the 3rd ACM International Conference on Multimedia*, 1995, pp. 231–236.

[18] R. B. Dannenberg, W. P. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis, "A comparative evaluation of search techniques for query-by-humming using the musart testbed," *Journal of the Association for Information Science and Technology*, vol. 8, no. 5, pp. 687–701, 2007.

[19] A. Kapur, M. Benning, and G. Tzanetakis, "Query-by-beating-boxing: Music retrieval for the DJ," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2004, pp. 170–177.

[20] O. Gillet and G. Richard, "Drum loops retrieval from spoken queries," *Journal of Intelligent Information Systems*, vol. 24, pp. 159–177, 2005.

[21] G. Roma and X. Serra, "Querying freesound with a microphone," in *Proc. the 1st Web Audio Conference (WAC)*, 2015.

[22] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams, "The timbre toolbox: Extracting audio descriptors from musical signal," *The Journal of the Acoustical Society of America*, vol. 130, no. 5, pp. 2902–2916, 2011.

[23] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[24] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transaction on*, vol. 26, no. 1, pp. 43–49, 1978.

[25] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on*, 2005, pp. 539–546.

[26] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. the 32nd International Conference on Machine Learning (ICML)*, 2015.

[27] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. European Conference on Computer Vision (ECCV)*, 2016, pp. 850–865.

[28] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *Proc. Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 2015, pp. 3279–3286.

[29] K. Chen and A. Salman, "Extracting speaker-specific information with a regularized siamese deep network," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2011, pp. 298–306.

[30] "Understanding and visualizing convolutional neural networks," http://cs231n.github.io/understanding-cnn/, accessed: 2017-09-30.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances in neural information processing systems (NIPS)*, 2012, pp. 1097–1105.

[32] D. Erhan, A. Courville, and Y. Bengio, "Understanding representations learned in deep architectures," *Department d'Informatique et Recherche Operationnelle, University of Montreal, QC, Canada, Tech. Rep 1355*, pp. 1–25, 2010.

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014, pp. 580–587.

[34] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. European conference on computer vision (ECCV)*, 2014, pp. 818–833.

[35] G. Montavon, "Deep learning for spoken language identification," in *Proc. NIPS Workshop on deep learning for Speech Recognition and Related Applications*, 2009, pp. 1–4.

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

[37] C. Schörkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *Proc. the 7th Sound and Music Computing Conference*, 2010.

[38] C. G. Gross, "Genealogy of the 'grandmother cell'," *The Neuroscientist*, vol. 8, no. 5, pp. 512–518, 2002.

[39] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. the 26th annual international conference on machine learning*, 2009, pp. 609–616.

[40] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[41] http://www.voxforge.org/ [Accessed 05/20/2018].

[42] D. R. Radev, H. Qi, H. Wu, and W. Fan, "Evaluating web-based question answering systems," *Ann Arbor*, vol. 1001, p. 48109, 2002.

[43] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[44] D. W. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *The Journal of Artificial Intelligence Research*, pp. 169–198, 1999.

[45] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, no. 2, pp. 236–243, 1984.

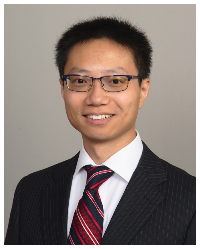
[46] T. Chi, P. Ru, and S. A. Shamma, "Multiresolution spectrotemporal analysis of complex sounds," *The Journal of the Acoustical Society of America*, no. 2, pp. 887–906, 2005.

[47] F. Pishdadian, B. Pardo, and A. Liutkus, "A multi-resolution approach to common fate-based audio separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017, pp. 566–570.

Yichi Zhang is a fourth-year Ph.D. candidate in the Department of Electrical and Computer Engineering at the University of Rochester, in the AIR Lab under the supervision of Prof. Zhiyao Duan. He received his M.S. degree in Optical Engineering focusing on optical fiber communications and DSP algorithms from Huazhong University of Science and Technology in 2014, under the supervision of Prof. Changjian Ke. He received his bachelors degree in Electrical and Information Engineering from Wuhan Univeristy of Technology in 2011. His research interests include machine learning, deep neural networks, and computer audition.

Bryan Pardo received an M. Mus. degree in jazz studies in 2001 and a Ph.D. degree in computer science in 2005, both from the University of Michigan, Ann Arbor. He is an associate professor in the Department of Electrical Engineering and Computer Science at Northwestern University, Evanston, IL. While finishing his doctorate he taught in the Music Department of Madonna University.

Dr. Pardo has authored over 80 peer-reviewed publications and he is an associate editor for IEEE Transactions on Audio Speech and Language Processing. When he is not programming, writing or teaching, he performs throughout the United States on saxophone and clarinet at venues such as Albion College, Chicago Cultural Center, Detroit Concert of Colors, Bloomington Indianas Lotus Festival, and Tucsons Rialto Theatre.



Zhiyao Duan is an assistant professor in the Department of Electrical and Computer Engineering at the University of Rochester, where he directs the Audio Information Research (AIR) laboratory. He also holds a secondary appointment in the Department of Computer Science and is affiliated with the Goergen Institute for Data Science. He received his B.S. in Automation and M.S. in Control Science and Engineering from Tsinghua University, China, in 2004 and 2008, respectively, and received his Ph.D. in Computer Science from Northwestern University in 2013. His research interest is in the broad area of computer audition, i.e., designing computational systems that are capable of understanding sounds, including music, speech, and environmental sounds.