2015 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2015, BOSTON, USA

RETRIEVING SOUNDS BY VOCAL IMITATION RECOGNITION

Yichi Zhang, Zhiyao Duan

Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627
{yichi.zhang, zhiyao.duan}@rochester.edu

ABSTRACT

Vocal imitation is widely used in human communication. In this paper, we propose an approach to automatically recognize the concept of a vocal imitation, and then retrieve sounds of this concept. Because different acoustic aspects (e.g., pitch, loudness, timbre) are emphasized in imitating different sounds, a key challenge in vocal imitation recognition is to extract appropriate features. Hand-crafted features may not work well for a large variety of imitations. Instead, we use a stacked auto-encoder to automatically learn features from a set of vocal imitations in an unsupervised way. Then, a multi-class SVM is trained for sound concepts of interest using their training imitations. Given a new vocal imitation of a sound concept of interest, our system can recognize its underlying concept and return it with a high rank among all concepts. Experiments show that our system significantly outperforms an MFCC-based comparison system in both classification and retrieval.

Index Terms— Sound retrieval, vocal imitation, automatic feature learning, stacked auto-encoder, multi-class classification

1. INTRODUCTION

Vocal imitation is widely used in human communication. To convey concepts of sounds that have a semantic meaning (e.g., dog barking), vocal imitation can help narrow down the concepts. For example, there are many kinds of dog barks, and vocal imitation can help distinguish infantile barks from “Christmas tree” barks. For sounds that do not have a definite semantic meaning (e.g., sounds from a synthesizer), vocal imitation is often the only way to convey the concepts.

Automatic vocal imitation recognition is a challenging but useful problem. It can augment current speech recognition technology to deal with non-speech vocalizations. It can also enable novel human-computer interactions. For example, current sound libraries are indexed and searched through semantic text labels. This approach, however, does not work for sounds that do not have a definite semantic meaning, such as synthesized sounds. These sounds are widely used and are constantly growing in the sound design industry. Experienced sound designers rely on their memory of the association between acoustic characteristics of the sounds and their production metadata to search for appropriate sounds. This expertise requires years of practice. Even for sounds with a semantic meaning, this approach would require users to memorize and differentiate many semantic labels. Take the previous dog barking example, if a user wants to search for a “Christmas tree” bark but does not know the name for it, he/she would have to listen through all kinds of dog barks before a wanted bark is found. Vocal imitation recognition, however, makes it possible to search for sounds that do not have a semantic meaning, and augments text-based search for sounds that have a definite semantic meaning.

A big challenge in vocal imitation recognition, however, is feature extraction. Vocal imitation conveys rich information covering many acoustic aspects: pitch, loudness, timbre, their temporal evolutions, and rhythmic patterns, etc. To imitate different sounds, people often imitate different aspects, those that mostly characterize the sound. For example, to imitate animal sounds like a cat meowing, both the pitch contour and timbre evolution play important roles, while for car horns, people often pay attention to the rhythmic pattern and timbre, but ignore the absolute pitch information. Identifying characteristic aspects for imitations is difficult. In some cases, imitators may even not be able to describe the aspect(s) they imitate. Therefore, finding features to represent these unclear aspects is very challenging.

In this paper, we propose a supervised approach to recognize vocal imitations. For each sound concept, we assume that a number of vocal imitations are available for training. A multi-class Support Vector Machine (SVM) is employed to learn to discriminate vocal imitations of different sound concepts. Then the classifier is able to classify a new vocal imitation to one of these trained sound concepts. For feature extraction, instead of using hand-crafted features, we employ an automatic feature learning approach to deal with the representation challenges described above. We use a stacked auto-encoder to learn features from a set of vocal imitations in an unsupervised way. Features learned in this way characterize acoustic aspects that human often imitate. To make the system more rigorous, we use different sound concepts to learn features and to construct the classifiers. Finally, we use the probability outputs of the multi-class SVM to rank classified sound concepts for sound retrieval.

Experiments are conducted on the VocalSketch Data Set v1.0.4 [1], which contains in total of 120 sounds and 4430 vocal imitations. We compare our approach with an MFCC-based multi-class SVM approach as a baseline. Results show that our approach significantly outperforms the baseline approach in both classification and retrieval.

In the following, we first review related work in Section 2, then describe the proposed approach in detail in Section 3. Experiments are presented in Section 4, and we conclude the paper in Section 5.

2. RELATED WORK

Broadly speaking, humming can be viewed as a vocal imitation of the melody of a song, and there exist extensive research on Query-by-Humming (QbH) in music information retrieval [2][3]. Existing systems for QbH almost all model pitch and inter-onset-interval variations, as they are the two exclusive aspects in which humming imitates the melody. The timbral aspect, for example, is not used in this imitation at all. This approach, however, does not work for recognizing and querying vocal imitations of general sounds.

Very few work has been done in studying vocal imitations of general sounds. Lemaitre et al. [4] tried to observe the relation between human discrimination of imitations and machine learning algorithm classification, using a general taxonomy of four branches as solid, liquid, gas, and electric in a kitchen scenario. However, the dataset they used was not large enough to cover commonly experienced sounds in our daily life. Blancas et al. [5] built a sound retrieval system by vocal imitation using temporal and spectral features and a SVM classifier. These hand-crafted features, we argue, are difficult to represent the complex acoustic aspects covered in a large variety of sound concepts and vocal imitations. In fact, their system was trained to distinguish only 3 or 4 sound concepts in each of 4 categories. Roma and Serra [6] built a Query-by-Example (QbE) system to query sounds from a large online database, but no formal evaluation has been conducted. Cartwright and Pardo [1] built a large vocal imitation dataset that covered four categories of sound concepts using Amazon's Mechanical Turk, but they did not propose a method for automatic vocal imitation recognition.

Another related but different concept from vocal imitation is onomatopoeia, and there exist some works in studying sound source selection [7] and music retrieval [8] by onomatopoeic queries, and the relationship between onomatopoeic representations and auditory impressions [9]. Both vocal imitation and onomatopoeia are ways for human to mimic a sound with voice. However, in vocal imitation, people could use all kinds of utterances to make the imitation livelier, as ventriloquist's performances are extreme examples. Onomatopoeia, on the other hand, has to use certain words from the language. Therefore, it is greatly restricted by the language and the culture, and is not necessarily linked to the actual acoustic content of the sound [10][11]. In this paper, we focus on recognizing and querying vocal imitation.

3. PROPOSED APPROACH

Figure 1 shows the four modules of the proposed system. For pre-processing, we represent each vocal imitation with a constant-Q spectrogram and segment it into short patches for further processing. We then use a trained stacked auto-encoder to extract features in each patch. Then a multi-class SVM is employed to classify each patch, and majority vote is conducted to obtain recording-level classifications. Finally, probabilistic classification outputs are used to rank sound concepts for sound retrieval. We discuss the details of each module in the following.

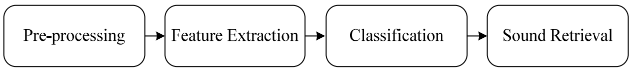


Figure 1. System overview

3.1. Pre-processing

Taking a monophonic vocal imitation as input, such as those in the VocalSketch dataset v1.0.4 [1] used in our experiment, the proposed system first downsamples it to 16 kHz. A 6-octave (50~3200 Hz) Constant-Q Transform (CQT) is then employed to calculate its spectrogram using the MATLAB CQT toolbox [12]. Each octave contains 12 bins and in total there are 72 frequency bins. The time frame hope size is 26.25 ms. We use CQT instead of short-time Fourier transform (STFT) because the log-frequency scale in CQT better corresponds to human auditory perception. In addition, the low frequency resolution at high frequencies of CQT provides a more compact representation of the imitation file for the next modules.

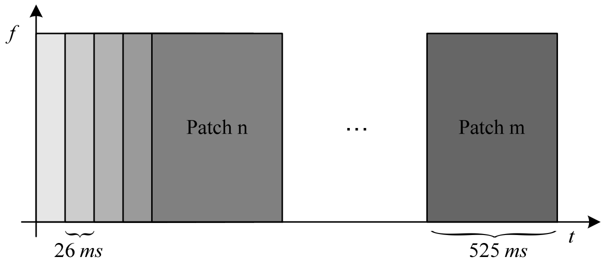


Figure 2. Patch segmentation of a CQT spectrogram

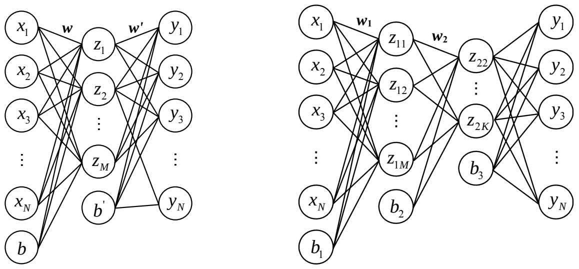
Lengths of vocal imitations vary significantly. If we were to represent each imitation file with a single feature vector of the same size, then if the size is too large, we will face the curse of dimensionality. But if the size is too small, we will miss too much detailed information of long files. So we need to decompose the CQT spectrogram into fixed-size short segments, or patches, and extract features in each patch. We set the length of each patch to be 20 frames (i.e., 525 ms). This value is chosen by considering the fact that in normal

English speech one syllable is about 250 ms long, which is the smallest unit to carry meaningful semantics. Therefore, each patch is represented by a 72*20 matrix, and is taken for feature extraction. Figure 2 illustrates the patch segmentation on the CQT spectrogram of an imitation sound file.

3.2. Feature Extraction

As discussed in Section 1, imitations of different sounds often attend to different acoustic aspects such as pitch, timber, loudness, and modulation, hence extracting appropriate features is a difficult problem.

Recently, features learned automatically by Deep Neural Networks (DNN) have shown significant advantages over deliberately handcrafted features in various tasks such as speech recognition and visual object detection. Features learned by DNN at different levels also show interesting hierarchical structures. For instance, Lee et al. [13] illustrated features extracted from human face images at different layers of a deep neural network model. Shallow layers detected local organs comprising the human face, while deeper layers extracted more holistic representations of the face.



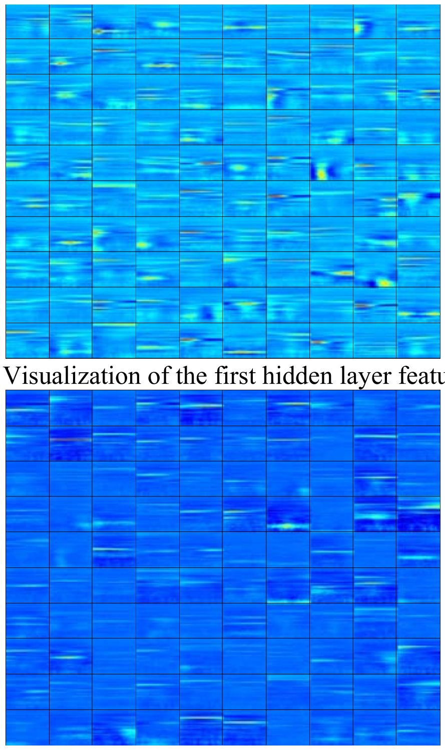
(a) (b)

Figure 3. Typical structures of (a) the auto-encoder and (b) the stacked auto-encoder

In this paper we choose to use *stacked auto-encoder* [14][15] for feature learning. Figure 3(a) shows a typical structure of an *auto-encoder* [14] with one hidden layer with untied weights. It is an unsupervised model. The transfer function of each hidden neuron and output neuron is a sigmoid function that squashes the input into a bounded output ranging from -1 to 1. This model tries to learn the parameters $\boldsymbol{w}$, b, $\boldsymbol{w'}$ and b', so that the output layer $\boldsymbol{y}$ approximates the input layer $\boldsymbol{x}$. Here $\boldsymbol{w}$ represents the weights between the input layer and the hidden layer, and $\boldsymbol{w'}$ represents weights between the hidden layer and the output layer. b and b' are the biases. If $M < N$, the hidden layer output is forced to learn a more compact representation of the input, which realizes dimension reduction.

If multiple auto-encoders are stacked together, we have the *stacked auto-encoder* [14], which is able to extract deep features. Figure 3(b) shows a stacked auto-encoder of two hidden layers. In this model, a greedy layer-wise training process is used to learn the parameters. Specifically, we begin with the training of the first hidden layer parameters $\boldsymbol{w_1}$, b_1, $\boldsymbol{w'_1}$ and b'_1 by feeding the imitation patches as input. $\boldsymbol{w'_1}$ and b'_1 are then discarded. Then we continue to train the second hidden layer parameters. To do so, we calculate the activation values of the first hidden layer, and treat them as inputs to the second hidden layer to learn $\boldsymbol{w_2}$, b_2, $\boldsymbol{w'_2}$ and b'_2. By following the same rule described above, we can keep moving with this mechanism if there are still more layers.

In our proposed approach, we adopt the two-hidden-layer stacked auto-encoder structure. In order to obtain the satisfying feature extraction performance, we set the number of neurons in the first and second hidden layers to 500 and 100, respectively. Weights connected from the previous layer to each neuron compose a feature. Therefore, there are 500 and 100 features in the first and second hidden layers, respectively. Figure 4 visualizes these features. Due to the limited space, we only display the first 100 out of 500 features in the first hidden layer. We can see that the first hidden layer extracts features that act as building blocks of the CQT spectrogram. The feature for each neuron in the second hidden layer is obtained by a weighted linear combination of features of the first hidden layer neurons to which it is strongly connected [16]. These features are more abstract.



(a) Visualization of the first hidden layer features

(b) Visualization of the second hidden layer features

Figure 4. Feature extraction visualization. Lighter color represents higher energy.

3.3. Multi-class Classification

Up to now each patch of the vocal imitation is represented by a 100-d feature vector. We then train a multi-class SVM using LIBSVM [17] to recognize their underlying sound concepts. For each sound concept, we assume that there are several vocal imitations available for training.

For a new vocal imitation whose underlying sound concept is unknown, the multi-class SVM classifies each patch of it to one of the trained sound concepts. Then majority vote is conducted to obtain the recording-level classification.

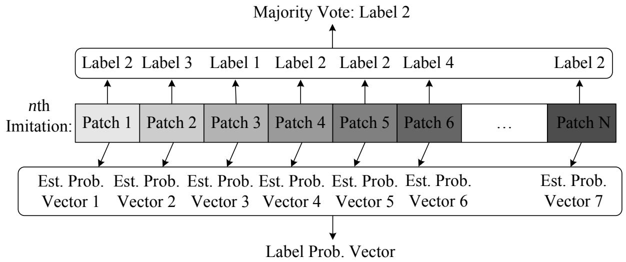


Figure 5. Illustration of recording-level classification calculation

3.4 Sound Retrieval

Given the classified sound concept, sounds of this concept can be retrieved. However, the returned concept may not always be correct. Therefore, in addition to the binary classification output, we also obtain a probabilistic classification output, showing the probability (confidence) that the vocal imitation patch belongs to each of the trained sound concepts. We then sort sound concepts according to their classification probabilities from high to low, and return sounds of highly-ranked concepts.

For sound concepts at the recording level, we average the probability output over all the patches in one imitation, and then sort sound concepts according to the averaged classification probability from high to low. Again, sounds of highly-ranked concepts can be retrieved.

Figure 5 illustrates the overall process, where the *n*-th imitation is comprised of a series of patches. Each patch has its own classification label and probability vector. The label appeared most frequently is chosen as the recording-level classification label. The probability vectors are averaged to obtain an average probability vector, based on which sound concepts are ranked.

4. EXPERIMENTAL RESULTS

4.1. Dataset

We use the VocalSketch Data Set v1.0.4 [1] in our experiments. This dataset contains sound concepts and their vocal imitation recordings in four categories: acoustic instruments, commercial synthesizers, everyday, and single synthesizer, which contain 40, 40, 120, and 40 sound concepts, respectively. For each sound concept, there are 20 to 40 vocal imitations from different people. The imitations were obtained through Amazon's Mechanical Turk. A detailed description of the sounds can be found in Table 1.

Table 1. Description of the VocalSketch v1.0.4 dataset [1]

Category	**Sound Concepts**
Acoustic instruments	Orchestral instruments playing a single note with the pitch C (in an appropriate octave chosen for each instrument)
Commercial synthesizers	Various recordings from Apple's Logic Pro music production suite
Everyday	A wide variety of acoustic events in everyday life
Single synthesizer	Recordings from a single 15-parameter subtractive synthesizer playing a note with the pitch C (octave varies depending on the parameter settings)

We use vocal imitations of the first half of all the sound concepts (ordered alphabetically) to train the stacked auto-encoder for feature learning, and use the second half to train and test the multi-class classifier within each category. This prevents the proposed system from over-fitting imitations that have been used for feature learning. Table 2 shows the number of sound concepts (i.e., classes) in each category used for feature learning and classification. It is noted that the single synthesizer category is not used in feature learning at all. For each sound concept for classification, we randomly choose 70% of the vocal imitations for training and the rest 30% for testing. In total there are 23,797 patches from 1,414 imitations for training and 9,767 patches from 601 imitations for testing.

Table 2. Number of sound concepts used for feature learning and classification

Category	**#Concepts for Feature Learning**	**#Concepts for Classification**
Acoustic instruments	13	17
Commercial synthesizers	17	13
Everyday	72	48
Single synthesizer	0	40

4.2. Evaluation Measures

We use two measures to evaluate the system performance: 1) classification accuracy, for vocal imitation classification; 2) Mean Reciprocal Rank (MRR), for sound concept retrieval. We calculate both measures at both the patch level and the recording level.

Classification accuracy is defined as the percentage of correctly classified imitations among all imitations. MRR is calculated as

$$MRR = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_i} \quad , \qquad (1)$$

where $rank_i$ is the rank of the correct sound concept in the probabilistic output of the i-th imitation patch or recording; Q is the total number of testing patches or recordings. MRR ranges from 0 to 1 with a higher value for a better performance. A value of 0.5 would suggest that the correct concept is ranked the 2nd among all concepts, on average.

4.3. Comparison Methods

In our proposed system the parameters are summarized as follows. For feature extraction, the CQT frame hop size is 26 ms. A patch contains 20 frames, hence is 525 ms long. The number of neurons in the first and second hidden layers of the stacked auto-encoder is 500 and 100, respectively. For the SVM classification, we use the Radial Basis Function (RBF) kernel, and tune the cost of constraints violation C = 1000 to get the highest classification accuracy.

We compare to a baseline system, which differs from the proposed system only at the feature extraction module. Instead of learning features automatically, the baseline system extracts MFCC features in each audio frame. The frame length is 52 ms and the hop size is 26 ms. Each MFCC feature vector is 39-d, containing 13 MFCC coefficients, 13 first-order time differences and 13 second-order time differences. This is a typical setting of feature extraction in many speech recognition and audio classification systems. We then view each frame as a patch and train a multi-class SVM for the classification. A RBF kernel is used and the cost of constraints violation parameter C is set to 100 to obtain the highest classification accuracy. Recording-level results are obtained in the same way as the proposed system. It is noted that this baseline system shares the same framework as [5]: SVM classification on frame-level hand-crafted features, although the features are not exactly the same.

4.4. Results

Table 3 and 4 show performance comparisons between the proposed system and the baseline method at the patch level and the recording level, respectively. Several interesting results can be observed:

First, both systems achieve significantly higher performance than random guesses at both the patch level and the recording level. Note that the random guess classification accuracies of the four categories would be 5.88%, 7.69%, 2.08%, and 2.50%, respectively. In Table 4, the highest MRR (0.4068) of the proposed system is obtained in the acoustic instruments category. This indicates that the correct sound concept is ranked between the 2nd and the 3rd among the 17 concepts in that category, on average. The lowest MRR (0.2581) is obtained in the everyday category. This value still tells that the correct sound concept is ranked the 4th among the 40 concepts in the category, on average. This indicates that the proposed supervised learning framework for vocal imitation recognition and retrieval is feasible and promising.

Table 3. Patch-level results

Category	Proposed		MFCC	
	Accuracy	MRR	Accuracy	MRR
Acoustic instruments	**19.54%**	**0.3519**	16.75%	0.3390
Commercial synthesizers	**14.73%**	**0.3052**	12.68%	0.2952
Everyday	**9.17%**	0.2043	8.30%	**0.2048**
Single synthesizer	**10.44%**	**0.2422**	7.64%	0.2114

Table 4. Recording-level results

Category	Proposed		MFCC	
	Accuracy	MRR	Accuracy	MRR
Acoustic instruments	**23.15%**	**0.4068**	19.44%	0.3578
Commercial synthesizers	**19.23%**	**0.3471**	12.82%	0.2890
Everyday	**9.49%**	**0.2581**	9.15%	0.2071
Single synthesizer	**12.50%**	**0.2830**	8.33%	0.2087

Second, the proposed system outperforms the baseline significantly at both levels in all categories except everyday. This supports our claim that features learned automatically are more suitable than hand-crafted features for vocal imitation recognition. One important reason for this is that temporal evolution plays an important role in vocal imitation. While the MFCC features can only model that across 3 frames, the automatically learned features are able to model temporal evolution within a patch, which contains 20 frames.

Third, by comparing Table 3 and 4, we can see that the recording-level classification and retrieval performance of both methods increase from their patch-level results. This increase is more significant for the proposed system in the categories of acoustic instruments and commercial synthesizers. This indicates that correct labels are more consistent than incorrect labels in the patch-level classification results, hence correct labels are more likely to be selected as recording-level labels after majority vote.

Finally, we compare performances in different categories. We can see that both systems achieve much better results in the acoustic instruments category than the commercial synthesizer category, although the former has 17 classes and the latter has only 13. After listening through all sounds and their imitations, we think that this is mainly because sounds in the acoustic instruments category are easier to imitate: they are all notes with a definite and static pitch. Sounds in the commercial synthesizer category, however, are more complex. Most of them contain multiple acoustic aspects such as transients, noise, and modulations on pitch and timbre. Therefore, they are more difficult to imitate, and less consistency is expected among different people's imitations. The everyday and single synthesizer categories have more classes, so the performance of both systems are lower.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an approach to automatically recognize the concept of a vocal imitation and retrieve sounds of this concept. A stacked auto-encoder model is adopted to learn features from a large variety of imitations in an unsupervised way. A multi-class SVM is trained to learn sound concepts using training imitations after feature extraction. It shows that our proposed system can recognize the underlying concept with a significantly higher-than-chance accuracy or return the correct concept with a high rank. The proposed system outperforms a baseline system that uses MFCC features in both classification and retrieval.

The biggest limitation of the current system is that it works in a supervised way, i.e., training vocal imitations are required to learn the concept of the sound. A future direction would be adopting unsupervised learning methods to generalize the system to retrieve sounds whose concepts are not trained. Besides, current features extracted by the staked auto-encoder are translation invariant along time but not frequency. This is problematic as imitations of the same sound (e.g., car horn) may use different pitches. We plan to make it translation invariant along frequency by segmenting the patches along the frequency axis as well. Finally, we would like to try better deep neural networks such as the Recurrent Neural Networks (RNN) to model the temporal evolution of imitations.

6. ACKNOWLEDGEMENT

We thank Mark Cartwright and Bryan Pardo for generously providing us with the VocalSketch Data Set v1.0.4.

7. REFERENCES

[1] Mark Cartwright and Bryan Pardo, "VocalSketch: Vocally Imitating Audio Concepts," in *Proc. ACM Conference on Human Factors in Computing Systems (CHI)*, 2015.

[2] Lie Lu, Hong You, and Hong-Jiang Zhang, "A New Approach to Query by Humming In Music Retrieval," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, pp. 152-155, 2001.

[3] M. Anand Raju, Bharat Sundaram, and Preeti Rao, "Tansen: A Query-By-Humming Based Music Retrieval System," in *Proc. National Conference on Communications (NCC)*, 2003.

[4] Guillaume Lemaitre, Arnaud Dessein, and Patrick Susini, "Vocal Imitations and the Identification of Sound Events," *Ecological Psychology*, Vol. 23, No. 4, pp. 267-307, 2011.

[5] David S. Blancas and Jordi Janer, "Sound Retrieval from Voice Imitation Queries in Collaborative Databases," in *Proc. AES 53rd International Conference on Semantic Audio*, London, pp.1-6, 2014.

[6] Gerard Roma and Xavier Serra, "Querying Freesound with a Microphone," in Proc. 1st Web Audio Conference (WAC), 2015.

[7] Yusuke Yamamura, Toru Takahashi, Tetsuya Ogata, et al., "Sound Source Selection System by Using Onomatopoeic Querries from Multiple Sound Sources," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2364-2369, 2012.

[8] Kenji Ishihara, Fuminori Kimura, and Akira Maeda, "Music Retrieval Using Onomatopoeic Query," in *Proc. World Congress on Engineering and Computer Science (WCECS)*, 2013.

[9] Masayuki Takada, Nozomu Fujisawa, Fumino Obata, and Shin-ichiro Iwamiya, "Comparisons of Auditory Impressions and Auditory Imagery Associated with Onomatopoeic Representation for Environmental Sounds," *EURASIP Journal on Audio, Speech, and Music Processing*, Vol. 2010, pp. 1-8, 2010.

[10] Shiva Sundaram and Shrikanth Narayanan, "Vector-based representation and clustering of audio using onomatopoeia words," in *Proc. The American Association for Artificial Intelligence (AAAI) Symposium Series*, Arlington, VA. 2006.

[11] Shiva Sundaram and Shrikanth Narayanan, "Classification of Sound Clips by Two Schemes: Using Onomatopoeia and Semantic Labels." in *Proc. IEEE International Conference on Multimedia and Expo*, pp. 1341-1344, 2008.

[12] Christian Schörkhuber and Anssi Klapuri, "Constant-Q transform toolbox for music processing," in *Proc. 7th Sound and Music Computing Conference*, Barcelona, Spain, pp. 3-64, 2010.

[13] Honglak Lee, Roger Grosse, Rajesh Ranganath, et al., "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical representations," in *Proc. The 26th International Conference on Machine Learning (ICML)*, Montreal, pp. 609-616, 2009.

[14] Andrew Y. Ng, "Sparse autoencoder," *CS294A Lecture notes 72*, pp. 1-19, 2011.

[15] Xiaojuan Jiang, Yinghua Zhang, Wensheng Zhang, et al., "A Novel Sparse Auto-Encoder for Deep Unsupervised Learning," *Sixth International Conference on Advanced Computational Intelligence*, Hangzhou, China, pp. 256-261, 2014.

[16] Honglak Lee, Chaitanya Ekanadham, and Andrew Y. Ng, "Sparse Deep Belief Net Model for Visual Area V2," in *Proc. Advances in Neural Information Processing System (NIPS)*, pp. 873-880, 2008.

[17] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM : A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, pp. 1-27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.