IMPROVING CONTENT-BASED AUDIO RETRIEVAL BY VOCAL IMITATION FEEDBACK

Bongjun Kim and Bryan Pardo

Northwestern University
Department of Electrical Engineering and Computer Science
Evanston, IL 60208, USA.

ABSTRACT

Content-based audio retrieval including query-by-example (QBE) and query-by-vocal imitation (QBV) is useful when search-relevant text labels for the audio are unavailable, or text labels do not sufficiently narrow the search. However, a single query example may not provide sufficient information to ensure the target sound(s) in the database are the most highly ranked. In this paper, we adapt an existing model for generating audio embeddings to create a state-of-the-art similarity measure for audio QBE and QBV. We then propose a new method to update search results when top-ranked items are not relevant: The user provides an additional vocal imitation to illustrate what they do or do *not* want in the search results. This imitation may either be of some portion of the initial query example, or of a top-ranked (but incorrect) search result. Results show that adding vocal imitation feedback improves initial retrieval results by a statistically significant amount.

Index Terms— Vocal imitation, content-based audio retrieval, relevance feedback, interactive information retrieval

1. INTRODUCTION

As the amount of multimedia content that includes audio (e.g. podcasts, music video, sound effects collections) increases, efficiently and accurately searching for desired audio content becomes increasingly important. Existing audio collections (e.g. SoundCloud, FreeSound, Youtube) are typically searched using text keywords associated with each audio file. However, text-based search fails when there is no text tag for the relevant audio content in the file. Text-based search is also not feasible when one seeks a sound that does not have a unique, commonly agreed-upon label known to the user (e.g. environmental sound from an unknown event).

A content-based audio retrieval system lets the user provide an audio example (e.g. a recording of a dog bark) as the query to find the desired audio (e.g. similar recordings of dogs barking). These systems ranks audio database items by their content similarity to the audio query example. Query-By-Example (QBE) [1, 2, 3, 4, 5, 6] and Query-By-Vocal Imitation (QBV) [7, 8, 9] are two kinds of content-based audio retrieval. QBE systems take an actual sound event (e.g. dog barking) as a query. QBV systems take a user's vocal imitation (e.g. an imitation of dog barking) as a query. QBV is useful when no recorded audio examples are available.

The performance of a QBE or QBV system depends on the quality of the query. If the query does not provide the right information to sufficiently narrow the search, the target sound(s) will not be top-ranked in the retrieval results. Getting user feedback indicating the relevance or irrelevance of a retrieved item is known as Relevance Feedback (RF) [10, 11, 12]. This feedback is used to help update the order of the returned search items. RF has been applied to text and image retrieval [13, 14]. Recently, it has also been applied to QBE [5, 6]. While there are previous works on both positive and negative relevance feedback in other domains [11, 12, 13, 14], we are unaware of work applying relevance feedback to QBV.

A simple approach to applying relevance feedback is to let a user label returned audio items in the search results as positive or negative. However, this does not make it clear what aspect of the audio item was positive or negative. In this work, we use vocal imitation as feedback. Vocally imitating sounds is a natural and effective way of communicating an audio concept between people [15]. Even when a vocal imitation sounds different from its original recording, it is often still identifiable [16]. We believe this is because it provides information about the temporal evolution of the target sound. This information can be used for relevance feedback to highlights what aspect of an example is positive or negative in a way that a simple label cannot.

In this work, we adapt an existing model for generating audio embeddings to create a state-of-the-art similarity measure for audio QBE and QBV. This model outperforms the current state-of-the-art QBV model for audio similarity measure [7]. We then show how to update both QBV and QBE retrieval results by a user's vocal imitation to provide feedback. We test our approach with multiple similarity models and show that information in feedback provided as a vocal imitation improves initial retrieval results by a statistically significant amount.

This work was supported by NSF Grant 1617497.

2. FEATURE EXTRACTION AND SIMILARITY MEASURE

A QBE or QBV system ranks recordings in the collection by similarity to the query (either a general audio sound or a vocal imitation). The current state-of-the-art QBV model, TL-IMINET [7] uses a Siamese style two-tower network that takes a vocal imitation as one input and an audio recording from the collection as the other, returning a similarity value for the pair. It was trained on a small number (thousands) of vocal imitations and takes fixed-length audio (a 4 second frame) as input. As a first step in our work, we decided to leverage the embeddings created from a much larger set of audio examples to build a better similarity measure that could easily handle arbitrary length recordings and also be useful for both QBE and QBV.

We use convolutional layers of *VGGish* model [17] that was trained on the roughly 3,000 sound classes in the YouTube-8M dataset [18] as a feature extractor. Figure 1(a) shows the model architecture. An audio file is first transformed into a log mel-spectrogram (64 Mel bins, a window size of 25 ms and hop size of 10 ms). This is passed to the trained *VGGish* model. The outputs from each intermediate layer in the model can be considered representations of the audio input. In this work, we use the last two convolutional layers, L5 and L6 which showed the best representation powers on a preliminary experiment. The outputs from these two layers are concatenated to form a feature vector for an audio example. Similarity between two sounds is calculated using cosine similarity between their feature vectors.

We tested three different ways of processing the outputs of L5 and L6 to create the final feature vector, as shown in Figure 1(b). Here, size is shown as (width, height, channel). The three variants are: 1)*VGGish-whole*: the network takes the whole length of audio (variable length). The size of the output from L5 or L6 is (t, 8, 512) where t depends on the length of input audio. Then, the output is averaged over t (average pooling) and flattened into a feature vector, 2) *VGGish-1s*: first, input audio is divided into a set of a non-overlapping 1-second segment. The network processes each segment (a spectrogram of size 96 by 64). Then, L5 or L6 output a matrix with a size of (12, 8, 512) and it is flattened into a feature vector without average pooling. To obtain a feature vector of a whole input recording, the network performs the same operation for every 1s segment of input audio and the output feature vectors are averaged across the segments. 3) *VGGish-2s*: this is the same as *VGGish-1s* except the network outputs a feature vector per 2s-segment (an input spectrogram of size 192 by 64) of input audio.

3. VOCAL IMITATION AS USER FEEDBACK

We present two scenarios where vocal imitation can help to improve a retrieval result, one for QBV and the other for QBE.

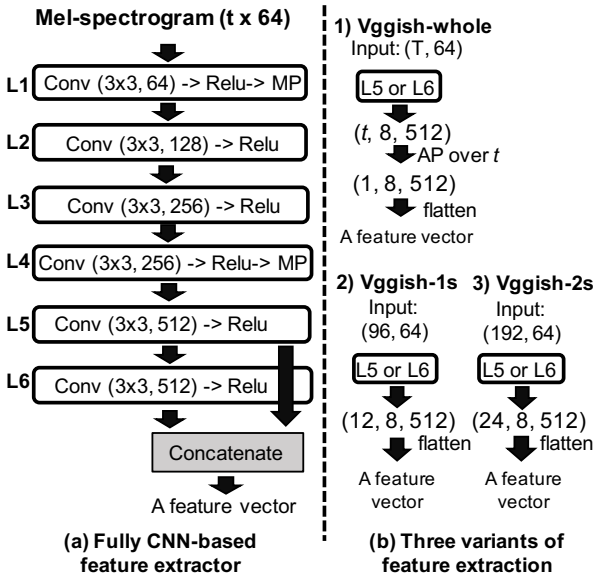

Fig. 1. (a) The proposed CNN-based feature extractor. The information for each filter is denoted as (width $\times$ height, channels) in each layer block. Outputs from layer 5 and 6 are used to build a feature vector. (b) Three variants of feature extractions.

First, in QBV, the initial query is a vocal imitation of the desired sound (i.e. a positive imitation). Here, one can provide feedback to the search results in two ways: either by providing an additional positive imitation or by imitating the top-ranked irrelevant sound item (a negative imitation), to highlight what aspect of that incorrect sound is not desired. The contrast between this negative vocal imitation and the original query (a positive vocal imitation) can be used to better guide search. The updated similarity S of x (a recording in the database) with q (a query) is computed as follows.

$$S(q,x) = \frac{1}{N_{vp}} \sum_{i=1}^{N_{vp}} C(v_p^i, x) - \frac{1}{N_{vn}} \sum_{i=1}^{N_{vn}} C(v_n^i, t) \cdot C(v_n^i, x) \quad (1)$$

Here, v_p^i is a i^{th} positive vocal imitation, N_{vp} is the total number of positive imitations (including an initial query), v_n^i is a i^{th} negative vocal imitation and N_{vn} is the number of negative imitations. C is a cosine similarity function. The term $C(v^i, t)$ determines how well a negative imitation captures the top-ranked erroneous search item t. If a user's imitation is very different from the original recording, the system puts a small weight on the imitation when computing the final similarity. This formula can flexibly accommodate multiple negative and positive imitations. One might ask why not just directly use the top-ranked incorrect example as a negative example. In our experiments we found the top-ranked erroneous example is less effective than a negative vocal imitation.

The second scenario we explore for using a vocal imitation for query refinement is a QBE task. Suppose a user has a

recording they wish to use as a query, but it contains multiple overlapping sounds. For example, if the query is a birdsong recorded in a natural setting, the example may also contain dog barking, lawn mowers, etc. In this case, a user can imitate the irrelevant sound event in the query (negative imitation) or imitate the relevant sound event in the query (positive imitation). We assume that the query has sound events overlapping each other: an event of interest and an irrelevant event. The system can flexibly accommodate one or more negative and positive imitations. The updated similarity S is computed as:

$$S(q,x) = C(q,x) - \frac{1}{N_{vn}} \sum_{i=1}^{N_{vn}} C(v_n^i, x) + \frac{1}{N_{vp}} \sum_{i=1}^{N_{vp}} C(v_p^i, x) \quad (2)$$

where v_n^i is a i^{th} vocal imitation of an irrelevant sound in the query (N_{vn} in total), and v_p^i is a i^{th} vocal imitations of a target sound in the query (N_{vp} in total).

4. EXPERIMENT-1: QUERY-BY-VOCAL IMITATION

To measure the performance gain from user's vocal imitation feedback on a QBV task, we tested our methods on two different QBV systems.

4.1. Dataset

We perform QBV retrieval on the VocalSketch dataset [16], which was used to train and test the current state-of-the-art QBV retrieval model [7]. It includes 240 reference recordings of 4 sub-categories: Acoustic Instruments (AI), Commercial Synthesizers (CS), Everyday Sound (ED), and Single Synthesizer (SS). Each sound recording in the dataset has at least 10 imitations collected through crowd-sourcing where the participants were asked to listen to reference audio recordings (e.g. the sound of a dog barking) and imitate them vocally. We have the same testing setup as [7]. There were 200 (AI), 204 (CS), 604 (ED), 204 (SS) queries, each performed on a dataset of 20 (AI), 20 (CS), 60 (ED), and 20 (SS) items.

4.2. Setting

To measure the performance gain by imitation feedback, we simulated a user's interaction with the search system as follows. A query (vocal imitation) in the testing set was selected. The reference recordings were ranked by the similarity with the query. If the top-ranked audio was not the target (the reference recording of the vocal imitation query), the initial search rankings were updated using an additional positive query or a vocal imitation of the top-ranked, but incorrect recording (negative example). Since there are 10 imitations per reference recording in the testing set, we randomly select one out of 10 imitations as user's negative feedback. To simulate user's multiple trials of a query or negative imitation, we picked different imitations of the same reference recording. Results were then re-ranked using Equation 1.

Table 1. MRRs on the VocalSketch testing set of 4 sub-categories of recordings *before* user feedback is submitted. (*TL-IMINET-paper: numbers from [7], TL-IMINET: our implementation)

Model	AI	CS	ED	SS
TL-IMINET [7]-paper	**0.462**	0.349	0.246	0.390
TL-IMINET [7]	0.454	0.334	0.235	0.402
VGGish-whole	0.409	0.363	0.292	0.422
VGGish-1s	0.429	0.378	0.329	0.436
VGGish-2s	0.459	**0.401**	**0.395**	**0.436**

We repeated this simulation over all the imitations (1,212) in the testing set. To measure performance, we computed Reciprocal Rank (RR) of the target reference recording as $RR = 1/rank$, where rank is the rank position of the reference recording. For example, RR is 0.25 if the target reference recording was retrieved at rank 4. The mean RR across all the queries is called the Mean Reciprocal Rank (MRR). To evaluate the general effect of vocal imitation feedback, we performed the simulation with 4 different models: TL-IMINET [7] which is the current state-of-the-art model and three variants of our model described in section 2: *VGGish-whole*, *VGGish-1s*, and *VGGish-2s*.

4.3. Results

We compare initial retrieval results (before user feedback is applied) from the current state-of-the-art QBV system, TL-IMINET [7] and three variants of our model. Table 1 shows within-category MRRs of the models. Since the trained model of TL-IMINET is not publicly available, we re-implemented it to test our feedback methods on it. Since our implementation achieved similar results to the original (see Table 1), we feel this provides a reasonable baseline. Compared with the baseline, even though our models were not trained on the VocalSketch dataset, every variant outperforms TL-IMINET for three of four categories. This result shows that audio classification models trained on a very large set of general sound events can be used as a feature extractor for QBV retrieval.

Table 2 shows how search results are improved by additional vocal imitations (one additional positive imitation and one negative imitation). We can see that vocal imitation feedback improves MRR on all the models for all the categories of the testing set, which confirms the general efficacy of vocal imitation feedback on QBV. Moreover, *VGGish-1s* and *VGGish-2s* outperformed TL-IMINET for all the categories after user's vocal imitation feedback is applied. The table also shows the performance gain only by either additional positive or negative imitation for *VGGish-2s* model. Both positive and negative vocal imitations contribute individually to improving retrieval. Using both improves MRR even more.

Interestingly, directly using the top-ranked incorrect

Table 2. MRRs updated by user's vocal imitation feedback and performance gain (Δ MRR) from the initial results in Table 1. (* indicates performance gain only by either additional positive (P) or negative (N) imitation)

Model	AI	CS	ED	SS
TL-IMINET [7]	0.503	0.439	0.259	0.452
TL-IMINET-Δ	0.049	0.105	0.025	0.050
VGGish-whole	0.475	0.457	0.382	0.482
VGGish-whole-Δ	0.066	0.094	0.090	0.059
VGGish-1s	0.512	0.457	0.429	0.505
VGGish-1s-Δ	0.082	0.079	0.101	0.070
VGGish-2s	0.545	0.488	0.499	0.518
VGGish-2s-Δ	0.087	0.087	0.104	0.083
*VGGish-2s-Δ (P)	0.051	0.066	0.091	0.063
*VGGish-2s-Δ (N)	0.058	0.031	0.020	0.023
VGGish-2s Neg-example	0.392	0.390	0.367	0.370
Negative-example-Δ	-0.064	-0.061	-0.031	-0.066

search result (not a vocal imitation of it) as the negative example hurts results for the *VGGish-2s* model (see *Negative-example-*Δ in Table 2), which confirms the effectiveness of a vocal imitation as negative feedback in QBV. We could not perform this test on TL-IMINET because it was trained to directly compare vocal imitations to general audio and was not trained to directly compare two general audio files.

5. EXPERIMENT-2: QUERY-BY-EXAMPLE

Now we evaluate the effectiveness of vocal imitation feedback when a query containing multiple overlapped sounds is provided to a QBE system.

5.1. Dataset

To test our QBE retrieval scenario, we need a dataset with multiple original recordings per class, which VocalSketch does not have. We used Vocal Imitation Set [19] which contains about 10 original recordings per class ($2,985$ original recordings of 302 classes in total) and one of them has about 20 vocal imitations collected through crowd-sourcing. The sound classes are curated from the AudioSet ontology [20]. The original recordings that have associated vocal imitations are called *reference* recordings and they are used as queries for this experiment. We ruled out one sound class which has only one original recording, which left us with 301 queries. All the other original recordings (2,683) that do not have vocal imitations become a set of items in a database that the model needs to search through to find target sounds. To simulate a *difficult query* situation, we create *mixed query* by mixing each of 301 queries (*clean query*) with another randomly selected query. Therefore, *clean query* has a single sound event and *mixed query* has overlapping sounds.

Table 3. Mean Recall@k for 6 types of queries. There were 301 queries, each performed on a set of 2,683 items. * indicates a statistically significant difference compared to results from MQ using a Wilcoxon signed-rank test ($p < 0.05$)

query type	MR@10	MR@20	MR@30
Clean Query (CQ)	0.265	0.356	0.412
Mixed Query (MQ)	0.128	0.172	0.211
MQ+Random	0.119	0.152	0.177
MQ+Positive	0.124	0.152	0.179
MQ+Negative	0.147*	0.187*	0.216*
MQ+Pos+Neg	0.144*	0.192*	0.225*

5.2. Settings and results

Our QBE system searches for target sounds almost in the same way as QBV does. The differences are that QBE takes an original recording as a query and the purpose is to find multiple target recordings (an average of 9 per query) that sound similar to the query, not to find a single target sound.

Given each of 301 mixed queries, the system returns a list of recordings in the testing set ordered by similarity with the query. If the *n* top-ranked recordings ($n = 5$ for this experiment) do not include any of target recordings, we augment the query using vocal imitations. We tested 6 different scenarios: 1) Clean Query only (CQ), 2) Mixed Query only (MQ), 3) MQ with Random imitation, 4) MQ with Positive imitation, 5) MQ with Negative imitation, and 6) MQ with Positive and Negative imitations. The random imitation is an imitation of a non-target recording in the testing set. Its purpose is to confirm that any random imitation is not helpful in improving retrieval results. We compute recall within top k items in search results (*Recall@*k). For example, Recall@20 of 0.7 means that 70% of target recordings are retrieved within top 20 items. We report the Mean Recall@k across all the queries.

Table 3 shows MeanRecall@k given different types of queries. To obtain the best upper-bound performance, we tested the 3 variants of our feature embedding models and picked the best model in MeanRecall@k given *clean query* which is *Vggish-whole* model. Results show that positive imitation was not helpful in this QBE scenario. Instead, negative imitation helps to improve the retrieval results significantly.

6. CONCLUSION

We presented a method to improve content-based audio retrieval using a user's vocal imitation feedback. We used a CNN-based feature extractor that takes a variable length of audio for both QBV and QBE retrieval. We also presented how to combine multiple similarities of a query and vocal imitation feedback with an item in the database. We showed that user's vocal imitation feedback improve the retrieval performance significantly.

7. REFERENCES

[1] Marko Helén and Tuomas Virtanen, "Query by example of audio signals using euclidean distance between gaussian mixture models," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, vol. 1, pp. I–225.

[2] Arefin Huq, Mark Cartwright, and Bryan Pardo, "Crowdsourcing a real-world on-line query by humming system," 2010.

[3] Ianis Lallemand, Diemo Schwarz, and Thierry Artières, "Content-based retrieval of environmental sounds by multiresolution analysis," in *SMC2012*, 2012, pp. 1–1.

[4] Jiachen Xue, Gordon Wichern, Harvey Thornburg, and Andreas Spanias, "Fast query by example of environmental sounds via robust and efficient cluster-based indexing," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 5–8.

[5] Bongjun Kim and Bryan Pardo, "I-sed: An interactive sound event detector," in *Proceedings of the 22Nd International Conference on Intelligent User Interfaces*, New York, NY, USA, 2017, IUI '17, pp. 553–557, ACM.

[6] Bongjun Kim and Bryan Pardo, "A human-in-the-loop system for sound event detection and annotation," *ACM Trans. Interact. Intell. Syst.*, vol. 8, no. 2, pp. 13:1–13:23, June 2018.

[7] Yichi Zhang and Zhiyao Duan, "Visualization and interpretation of siamese style convolutional neural networks for sound search by vocal imitation," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[8] Adib Mehrabi, Simon Dixon, and Mark B Sandler, "Vocal imitation of synthesised sounds varying in pitch, loudness and spectral centroid," *The Journal of the Acoustical Society of America*, vol. 141, no. 2, pp. 783–796, 2017.

[9] Mark Cartwright and Bryan Pardo, "Synthassist: an audio synthesizer programmed with vocal imitation," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 741–742.

[10] Yong Rui, Thomas S Huang, Michael Ortega, and Sharad Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Transactions on circuits and systems for video technology*, vol. 8, no. 5, pp. 644–655, 1998.

[11] Maryam Karimzadehgan and ChengXiang Zhai, "Improving retrieval accuracy of difficult queries through generalizing negative document language models," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 27–36.

[12] Yunlong Ma and Hongfei Lin, "A multiple relevance feedback strategy with positive and negative models," *PloS one*, vol. 9, no. 8, pp. e104707, 2014.

[13] Bart Thomee and Michael S Lew, "Interactive search in image retrieval: a survey," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 2, pp. 71–86, 2012.

[14] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder, "Cueflik: Interactive concept learning in image search," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2008, CHI '08, pp. 29–38, ACM.

[15] Guillaume Lemaitre and Davide Rocchesso, "On the effectiveness of vocal imitations and verbal descriptions of sounds," *The Journal of the Acoustical Society of America*, vol. 135, no. 2, pp. 862–873, 2014.

[16] Mark Cartwright and Bryan Pardo, "Vocalsketch: Vocally imitating audio concepts," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 43–46.

[17] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al., "Cnn architectures for large-scale audio classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 131–135.

[18] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[19] Bongjun Kim, Madhav Ghei, Bryan Pardo, and Zhiyao Duan, "Vocal imitation set: a dataset of vocally imitated sound events using the audioset ontology," in *Proceedings of the 2018 Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE2018)*, 2018.

[20] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 776–780.