High-performance Audio Matching with Features Learned by Convolutional Deep Belief Network

Weijiang Feng[1], Naiyang Guan[1], Zhigang Luo[1,2]
[1]Institute of Software, College of Computer
National University of Defense Technology, Changsha, Hunan, P.R. China, 410073
[2]Science and Technology on Parallel and Distributed Processing Laboratory
National University of Defense Technology, Changsha, Hunan, P.R. China, 410073
Email: wjfeng1992@163.com, ny_guan@nudt.edu.cn, zgluo@nudt.edu.cn

***Abstract*—Audio matching automatically retrieves all excerpts that have the same content as the query audio clip from given audio recordings. The extracted feature is critical for audio matching and the Chroma Energy Normalized Statistics (CENS) feature is the state-of-the-arts. However, CENS might behave unsatisfactorily on some audio because it is a handcraft feature. In this paper, we propose to utilize the features learned by Convolutional Deep Belief Network (CDBN) to enhance the performance of audio matching. Benefit from the strong generalization ability of CDBN, our method works better than CENS based methods on most audio datasets. Since the features learned by CDBN are binary-valued, we can develop a more efficient audio matching algorithm by taking the advantage of this property. Experimental results on both TIMIT dataset and a simulated music dataset confirm effectiveness of the proposed CDBN based method comparing with the traditional CENS feature based algorithm.**

Keywords: audio matching; convolutional deep belief network; content-based audio retrieval

I. Introduction

Content-based audio retrieval and analysis is a challenging problem in signal processing. A large amount of attention has been paid to the query-by-example paradigm in this literature, namely, given an audio recording or an excerpt of it (or called query clip), the query-by-example paradigm automatically retrieves segments having similar content to the query clip from a given audio database [1]. Since the waveforms in the audio data are large-scale and noisy, the content-based audio retrieval is quite challenging especially on the digital waveform-based audio data.

In the content-based audio retrieval literature, audio matching is an important problem which aims to retrieve all excerpts from all recordings within a given audio database so that all retrieved audio excerpts in some sense represent the same content as the query audio clip [1], [2]. A typical scenario for audio matching is when the same piece of song (in an abstract sense) is available in several specific interpretations. For example, given a ten-second excerpt of Queen's interpretation of the song "we will rock you", the goal is to find all corresponding audio clips in the database, and these clips include the repetition in the exposition or in the recapitulation within the same interpretation as well as the corresponding excerpts in all recordings of the same song interpreted by other singers such as "Britney Spears", "Russian Red", and "The Park".

Traditional audio matching depends on the results of audio identification which identifies the audio recording containing the query one from the audio database. Currently, audio identification algorithms show a significant progress even in the presence of noise, compression artifacts, and slight temporal distortions of the query [3], [4]. However, these algorithms cannot deal with strong nonlinear temporal distortions or with other variations that concern, e.g., the articulation or spectral deviations [2]. Recently, Frank Kurth and Meinard Muller [1], [2] proposed Chroma Energy Normalized Statistics (CENS) feature for audio matching. CENS temporally blurred chroma features to achieve robustness to local tempo variations first. Then CENS normalized the resulting features to achieve invariance to deviations in dynamics [2]. The CENS feature shows a high degree of robustness to variations in dynamics, timbre, articulation, and local tempo deviations. However, the CENS feature is handcrafted, thus has a poor generalization performance.

In this paper, to overcome the shortcomings of CENS, we resort to unsupervised feature learning for audio matching. We employed convolutional deep belief network (CDBN) [5], [6] to automatically extract unsupervised features from audio data. It turns out that the audio features extracted by CDBN are robust to noise, and are discriminative to distinguish unrelated audio clips. Furthermore, based on the original audio matching procedure, we proposed a more efficient audio matching algorithm dedicated to binary-valued features extracted by CDBN. We evaluated our audio matching algorithm based on features learned by CDBN on TIMIT dataset and a simulated music dataset. Experimental results confirm the effectiveness of the proposed algorithm.

II. Convolutional Deep Belief Network based Audio Matching

The whole procedure of the proposed audio matching method is illustrated by Figure 1. First, the audio data are pre-processed, the details of which are described in section III-A. Then, we apply the convolutional deep belief network (CDBN) to extract features from audio data. The CDBN is built by stacking two convolutional restricted Boltzmann

machine (CRBM), and the principal of CRBM is explained in section II-A. Finally, audio matching is conducted based on the features learned by CDBN, and section II-B specifies the CDBN based audio matching algorithm.

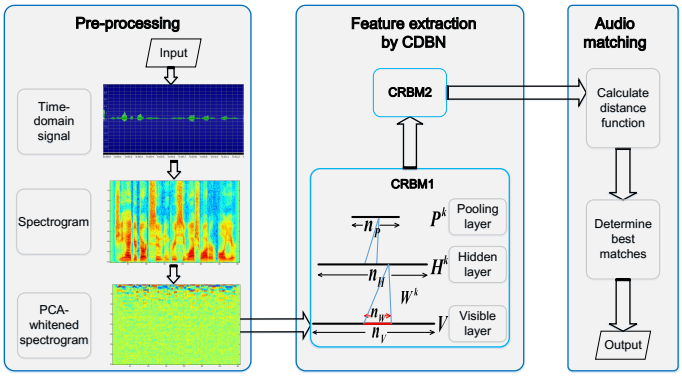

Fig. 1. The whole procedure of our audio matching method.

A. Feature Extraction by CDBN

Feature extraction plays a critical role in audio matching, however, conventional audio matching methods extract hand-crafted features and thus have weak generalization ability. Deep learning can automatically learn features and has been widely used in various applications such as image classification [7]–[9], object detection and segmentation [10], and speech recognition [11] due to its good generalization ability.

In the deep learning literature, convolutional deep belief network (CDBN) [5], [6] takes the advantages of both deep belief network (DBN) [12] and convolutional neural network (CNN) [7]–[10], and can scale up the unsupervised DBN on high-dimensional data. CDBN has shown pleasant performance in visual recognition tasks [5] and audio classification [6]. We therefore apply CDBN to unlabeled auditory data to extract features for audio matching.

We first describe the elementary component of CDBN, namely convolutional restricted Boltzmann machine (CRBM) [5]. Following [6], we assume that all inputs are single-channel time-series data with n_V frames, namely a n_V-dimensional vector, and such formulation can be straightforwardly extended to multiple channels.

Intuitively, CRBM consists of two layers including the visible layer V and the hidden layer H. The visible units are binary-valued or real-valued, while the hidden units are restricted to be binary-valued. Weights between the visible units and the hidden units are shared among all locations in the hidden layer. Assuming that the visible layer accepts n_V-dimensional vector of binary values as input, and contains totally K feature detectors $W^K \in R^{nw}$. The hidden layer consists of K groups of n_H-dimensional vectors, where $n_H = n_V - n_W + 1$, and the units in group k share the weights W^k. Besides, there is a shared bias b_k for each unit in hidden group k, and a shared bias c for visible units.

Given the visible units v, the hidden units h are computed according to one conditional probability as follows:

$$P\left(h_j^k = 1|v\right) = sigmoid\left(\left(\tilde{W}^k *_v v\right)_j + b_k\right) \quad (1)$$

After computing the hidden units h, we reconstruct the visible units $\tilde{v}$ according to the following conditional probabilities:

$$P\left(\tilde{v}_i = 1|h\right) = sigmoid\left(\sum_k \left(W^k *_f h^k\right)_i + c\right) \quad (2)$$

$$P\left(\tilde{v}_i|h\right) = Normal\left(\sum_k \left(W^k *_f h^k\right)_i + c, 1\right), \quad (3)$$

where $*_v$ is called valid convolution, $*_f$ is called full convolution, $\tilde{W}_j^k \triangleq W_{n_W-j+1}^k$. Equation (2) is for binary visible units, while equation (3) is for real visible units. Following [5], we also uses a probabilistic max-pooling layer, where the maxima over small neighborhoods of hidden units are computed. The pooling layer contains n_P-dimensional vectors.

CRBM with probabilistic max-pooling are the building blocks for CDBN. By stacking max-pooling-CRBM on top of one another, we define the architecture of CDBN, and this is analogous to the construction of DBN. Training of the CDBN is also layer-wise: once the training of a given layer is accomplished, the corresponding weights remain fixed, and the activations are fed into the next layer as its input. For each input audio, the extracted audio features are binary-valued, while CENS outputs real-valued features.

B. CDBN Based Audio Matching

Based on the features learned by CDBN, we describe a novel audio matching algorithm under the framework of [1]. The audio database consists of a collection of audio recordings, with some audio being the same sentence spoken by different individuals. We represent the audio database by one large document D by concatenating all individual recordings, and the short query audio clip Q. All the audio recordings and the query audio clips have C channels. We define $\Omega_C := \left\{x = (x_1, \cdots, x_C)^T | x_i \in \{0,1\},\ i = 1:C\right\}$. In the feature extraction step, both document D and the query Q are transformed into binary-valued feature sequences. We denote these feature sequences by $F[D] = \left(v^1, \cdots, v^N\right)$ and $F[Q] = \left(w^1, \cdots, w^M\right)$ with $v^n \in \Omega_C$ for $n \in [1:N]$ and $w^m \in \Omega_C$ for $m \in [1:M]$.

The goal of audio matching is to identify audio clips in D whose content are the same as that of Q. We compare the feature sequence $F[Q]$ to any subsequence $F[D]$ consisting of M consecutive vectors. Specifically, letting $X = \left(x^1, \cdots, x^M\right) \in \Omega_C^M$, $Y = \left(y^1, \cdots, y^M\right) \in \Omega_C^M$ and defining $\Theta(x,y) = \frac{1}{C}\sum_{i=1}^{C} \sigma(x_i, y_i)$ with $x \in \Omega_C, y \in \Omega_C$ under the definition of σ as follows:

$$\sigma(s,t) = \begin{cases} 1 & \text{if } s = t \\ 0 & \text{else.} \end{cases} \quad (4)$$

Then we set $d^M(X,Y) := 1 - \frac{1}{M}\sum_{m=1}^{M}\Theta(x^m, y^m)$ with $X \in \Omega_C^M, Y \in \Omega_C^M$. Note that d^M is in the real interval $[0,1] \subset \Re$ and equals to zero if X and Y coincide. Next, with respect to $F[D]$ and $F[Q]$, we define the distance function $\Delta : [1:N] \to [0,1]$ by $\Delta(i) := d^M\left(\left(v^i, v^{i+1}\cdots, v^{i+M-1}\right), \left(w^1, w^2\cdots, w^M\right)\right)$ for $i \in [1:N-M+1]$ and $\Delta(i) := 1$ for $i \in [N-M+2:N]$. In particular, $\Delta(i)$ describes the distance between $F[Q]$ and one subsequence of $F[D]$, which consists of M consecutive vectors and starts at position i.

The computation of Δ can be illustrated in Fig. 2, which slides the window $F[D]$ of size M with the step size 1 to compare with the query clip $F[Q]$.

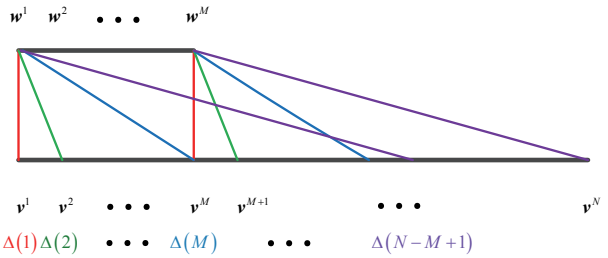

Fig. 2. Illustration of the computation of the distance function Δ with respect to $F[Q] = \left(w^1, \cdots, w^M\right)$ and $F[D] = \left(v^1, \cdots, v^N\right)$.

The best matches of Q within D is determined by successively selecting minima of the distance function Δ: in the first step, the index $i \in [1:N]$ minimizing Δ is determined, indicating that the audio clip corresponding to the feature sequence $\left(v^i, \cdots, v^{i+M-1}\right)$ is the best match. Then a neighborhood of length M of the best match is excluded for further considerations to avoid subsequent matches with a large overlap to the previous best match, by setting $\Delta(j) = 1$ for $j \in [i - \lceil M/2 \rceil : i + \lceil M/2 \rceil] \cap [1:N]$. In the second step, the feature index minimizing the modified distance function is determined, resulting in the second best match. This procedure will not stop until we have retrieved a predefined number of matches or a retrieved match exceeds a specific threshold.

The primary cost of our matching algorithm is the computation on the comparison of two vectors, i.e., $\Theta(x,y)$, which is more efficient than the inner product of two vectors [1]. Therefore, the proposed audio matching algorithm is faster than the original audio matching algorithm [1].

III. EXPERIMENTS

We implemented our CDBN based audio matching algorithm in MATLAB and evaluated it on the TIMIT dataset and a simulated music dataset consisting of songs collected from the Internet.

A. Training CDBN on the TIMIT Dataset

Following the guideline [6], we first convert time-domain signals into spectrograms, which have a 20 ms window size with 10 ms overlaps. Then, we applied PCA whitening (with 80 components) to decrease the dimensionality of the spectrograms. As a result, the data we fed into the CDBN consists of 80 channels of one-dimensional vectors.

Our CDBN has two hidden layers. To train the network, we use a large, unlabeled speech dataset TIMIT [13]. We trained 300 first-layer feature detectors with a filter length (n_W) of 6 and a max-pooling ratio (local neighborhood size) of 3. We further trained 300 second-layer feature detectors using the max-pooled first-layer activations as input, again with a filter length of 6 and a max-pooling ratio of 3. The first layer was trained for 500 iterations, and the second layer was trained for 200 iterations. What the network has learned can be illustrated through visualization. We visualize the first-layer feature detectors by multiplying the inverse of the PCA whitening on each feature detector, as shown in Figure 3. The feature detectors have learned different audio features. Some feature detectors take the form of energy distributed in the high frequencies of the spectrogram, some in the low frequencies, and some uniformly in the whole frequency band.

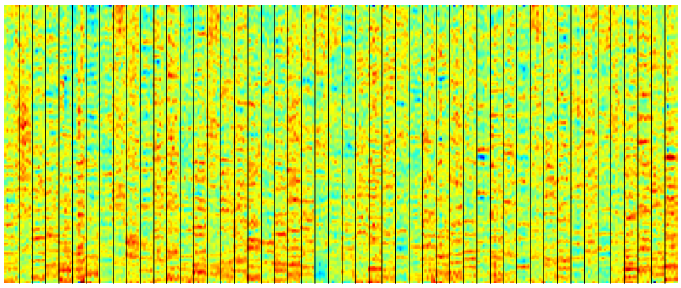

Fig. 3. Visualization of randomly selected first-layer CDBN feature detectors trained on the TIMIT data in the spectrogram space.

B. Audio Matching Results on the TIMIT Dataset

TIMIT contains a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions (dr1-dr8) of the United States. In each dialect region, there exist 2 sentences (sa1, sa2) spoken by more than 20 individuals, much more sentences spoken by only one individual (si–), and some sentences spoken by one to five individuals (sx–). In each dialect region, we concatenate all the small audio files into one large audio file, obtaining 8 concatenated audio documents, one for each dialect region. We select 8 short audio files (sa1, sa2, 3 si–, and 3 sx–) as query files in each dialect region, and query them in the corresponding concatenated audio document. The selected query files in each dialect region and the number of repetitions of these query files in corresponding concatenated audio document are indicated in Table I.

We now discuss in detail the matching results obtained from our audio matching algorithm. For a query having n_q repetitions in the database, the audio matching procedure will return n_q indexes indicating predicted positions of the query's repetitions. If the predicted position is exactly the position of the query's repetition, then it is a "hit". The hit number of matching for query audio files are shown in Figure 4. Apart

TABLE I
THE QUERY AUDIO FILE NAME AND THE NUMBER OF REPETITIONS FOR EACH QUERY FILE IN CORRESPONDING DIALECT REGION.

dr	query file name, number of repetition							
dr1	sa1, 38	sa2, 38	si518, 1	si1025, 1	si2004, 1	sx22, 2	sx114, 3	sx451, 2
dr2	sa1, 76	sa2, 76	si528, 1	si1008, 1	si2001, 1	sx6, 3	sx86, 4	sx446, 4
dr3	sa1, 76	sa2, 76	si454, 1	si1006, 1	si2010, 1	sx33, 5	sx123, 5	sx393, 5
dr4	sa1, 68	sa2, 68	si457, 1	si1034, 1	si2015, 1	sx79, 3	sx259, 3	sx439, 3
dr5	sa1, 70	sa2, 70	si468, 1	si1008, 1	si2006, 1	sx16, 3	sx60, 3	sx81, 3
dr6	sa1, 35	sa2, 35	si474, 1	si1048, 1	si2012, 1	sx7, 2	sx70, 2	sx160, 2
dr7	sa1, 77	sa2, 77	si487, 1	si1005, 1	si2011, 1	sx17, 4	sx39, 3	sx59, 3
dr8	sa1, 22	sa2, 22	si486, 1	si1044, 1	si2018, 1	sx3, 3	sx172, 3	sx442, 3

from results obtained based on CENS [1] features, we also compare our results with those obtained based on baseline MFCC features. Here, "CDBN L1" means layer1 features, "CDBN L2" means layer2 features.

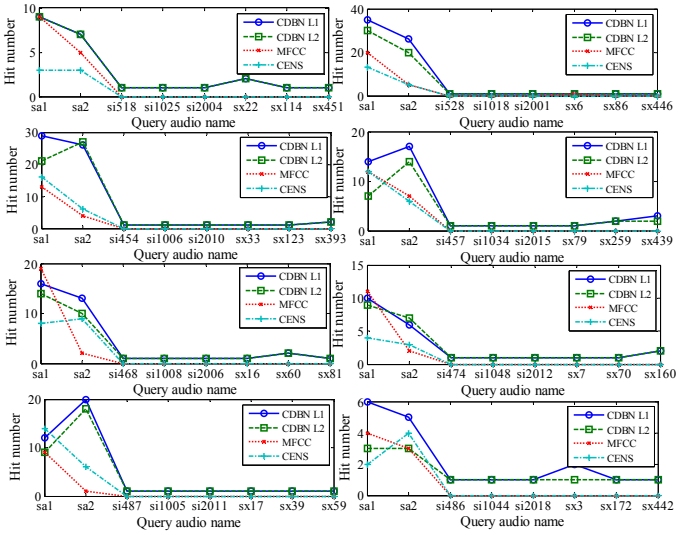

Fig. 4. The hit number of audio matching for query audio files with CDBN L1, CDBN L2, MFCC, CENS features. Each sub-figure illustrates one dialect region: the left top dr1, the right top dr2, the right bottom dr8 and so on.

From Figure 4, it is clear that our audio matching procedure far outperforms the MFCC and CENS feature based matching procedure, no matter for query audio files having more than 20 repetitions in the database or for queries that have limited repetitions. Specifically, for "si–" query audio files, our audio matching procedure successfully locate all these query files in corresponding concatenated audio document, while MFCC and CENS features based methods fail to locate any one of these query files; for "sx–" query audio files, our method locate a part of repetitions of these query files, while MFCC and CENS features based methods locate none of the repetitions of these query files; and for sa1 and sa2 query audio files, our method successfully locate more repetitions for most of these query audio files.

To verify the robustness to noise of our audio matching method, we add white Gaussian noise to these query audio files, obtaining three extra audio files for each query, and the SNR of these three extra audio files are 10db, 20db, and 30db, respectively. The hit number of the original clean query audio files and the corrupted audio files by white Gaussian noise are pictured in Figure 5 and Figure 6. In Figure 5, the results are obtained using CDBN layer1 features, and in Figure 6, we use CDBN layer2 features.

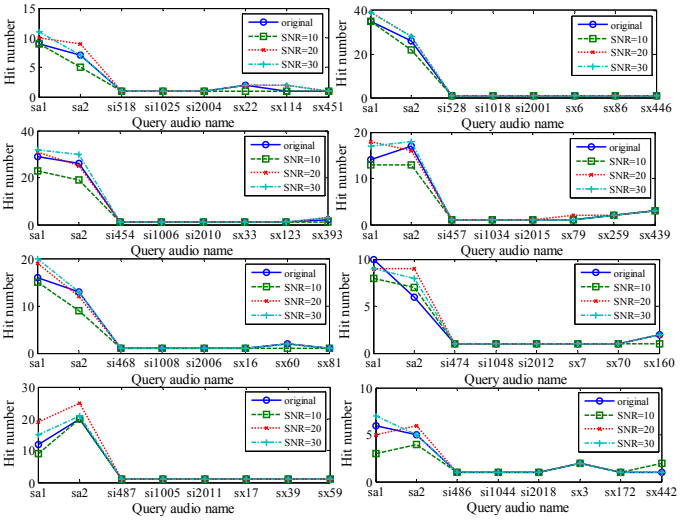

Fig. 5. The hit number of audio matching for original clean query audio files and corrupted query files with CDBN layer1 features. Each sub-figure illustrates one dialect region: the left top dr1, the right top dr2, the right bottom dr8 and so on.

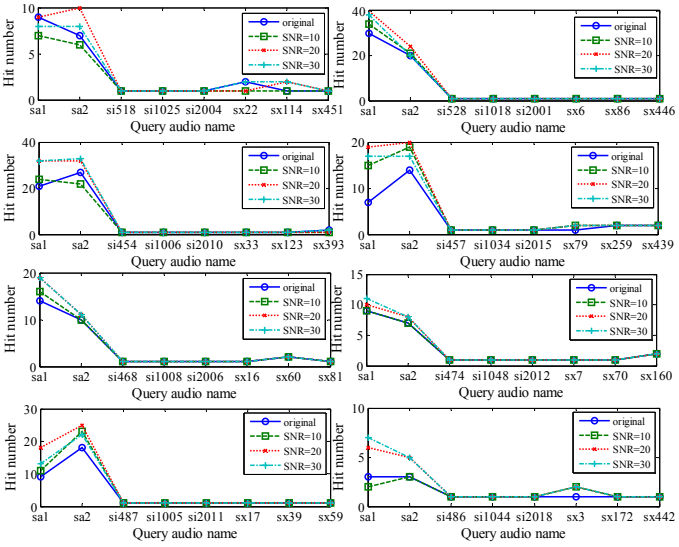

Fig. 6. The hit number of audio matching for original clean query audio files and corrupted query files with CDBN layer2 features. Each sub-figure illustrates one dialect region: the left top dr1, the right top dr2, the right bottom dr8 and so on.

From Figure 5 and Figure 6, we find that the hit number of the corrupted query audio files are larger than or equal to that of the corresponding original clean query audio for most audio files using both CDBN layer1 features and CDBN layer2

TABLE II
THE HIT NUMBER OF AUDIO MATCHING FOR QUERY AUDIO CLIPS WITH DIFFERENT AUDIO FEATURES.

query audio clip	($leng\_t$, #)	MFCC	CENS	CDBN L1	CDBN L2
Better Man	(3s, 17)	0	0	**8**	7
	(9s, 12)	1	2	**6**	**6**
	(20s, 12)	0	0	**9**	8
God is A Girl	(2s, 21)	1	5	**18**	17
	(7s, 5)	0	1	**4**	**4**
	(30s, 5)	0	1	**5**	**5**
Halo	(3s, 8)	2	0	**4**	3
	(17s, 8)	3	1	**5**	**5**
	(29s, 6)	0	0	**3**	**3**
Rolling in the Deep	(4s, 18)	1	1	**10**	**10**
	(9s, 18)	1	2	**11**	**11**
	(19s, 18)	6	2	14	**16**
We will Rock You	(4s, 40)	13	8	**32**	31
	(10s, 19)	7	4	**10**	**10**
	(23s, 4)	0	1	**2**	**2**
Yesterday Once More	(7s, 10)	2	0	**7**	**7**
	(10s, 14)	2	0	**10**	9
	(21s, 12)	1	2	**10**	9

features, demonstrating the robustness to noise of our audio matching procedure.

C. Audio Matching Results on Music Dataset

We also verify the effectiveness of our method using a simulated music dataset. The music dataset, collected from the Internet, consists of these songs: "Better man", "God is a girl", "Halo", "Rolling in the deep", "We will rock you", and "Yesterday once more", each of these songs is performed by several different singers. For each piece of song, there are three query audio clips with different length of time. The longest query among the three queries for all pieces of songs is from 20s to 30s, the shortest query is between 2s and 7s, and the middle is from 7s to 17s. The hit number of matching for query audio clips are shown in Table II. Here, each querys duration time ($leng\_t$) and the number of repetitions (#) in the dataset are given in the second column. The song, from which the query audio clip is extracted is given in the first column. Table II again demonstrates that our audio matching procedure far outperforms the MFCC and CENS features based audio matching method.

IV. CONCLUSION

In this paper, we propose a novel audio matching algorithm based on features learned by CDBN. Specifically, given a query audio clip, one should automatically and efficiently identify all audio segments having the same content as the query clip in the database. We apply convolutional deep belief network (CDBN) to extract audio features, and propose an efficient matching algorithm based on the extracted audio features. Experimental results on TIMIT dataset and a simulated music dataset collected by ourselves from the Internet demonstrated that the proposed audio matching algorithm significantly outperforms the MFCC and CENS features based audio matching algorithm, and that the extracted features by CDBN are robust to Gaussian noise.

REFERENCES

[1] M. Müller, F. Kurth, and M. Clausen, "Audio matching via chroma-based statistical features." in *ISMIR*, vol. 2005, 2005, p. 6th.

[2] F. Kurth and M. Müller, "Efficient index-based audio matching," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 382–395, 2008.

[3] E. Allamanche, J. Herre, O. Hellmuth, B. Froba, and M. Cremer, "Audioid: Towards content-based identification of audio material," *Preprints-Audio Engineering Society*, 2001.

[4] P. Cano, E. Batle, T. Kalker, and J. Haitsma, "A review of algorithms for audio fingerprinting," in *Multimedia Signal Processing, 2002 IEEE Workshop on*. IEEE, 2002, pp. 169–173.

[5] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 609–616.

[6] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

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

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *arXiv preprint arXiv:1502.01852*, 2015.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

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

[11] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[13] W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall, "The darpa speech recognition research database: specifications and status," in *Proc. DARPA Workshop on speech recognition*, 1986, pp. 93–99.