Musical Query-by-Semantic-Description Based on Convolutional Neural Network

Jing Qin[1,2], Hongfei Lin[1(✉)], Dongyu Zhang[1], Shaowu Zhang[1], and Xiaocong Wei[3]

[1] School of Computer Science and Technology, Dalian University of Technology, Dalian, China
qinjing@dlu.edu.cn, hflin@dlut.edu.cn
[2] College of Information Engineering, Dalian University, Dalian 116622, China
[3] School of Software Engineering, Dalian University of Foreign Language, Dalian 116044, China

Abstract. We present a new music retrieval system based on query by semantic description (QBSD) system, by which a novel song can be used as query and transformed into semantic vector by a convolutional neural network. This method based on Supervised Multi-class labeling (SML), which a song can be annotated by some semantically meaningful tags and retrieved relevant song in semantically annotated database. CAL500 data set is used in experiment, we can learn a deep learning model for each tag in semantic space. To improve the annotation effect, loss function adjustment algorithm and SMOTE algorithm are employed. The experiment results show that this model can get songs with high semantically similarity, and provide a more nature way to music retrieval.

Keywords: Query by semantic description · Convolutional neural network · Supervised multi-class labeling · Semantically retrieval

1 Introduction

As the size of audio and music collection dramatically increase in internet, millions of songs are available to consumers online and it is difficult to find and discover new songs. In music industry, the music retrieval systems mainly used manual retrieval [1], which based on mete-data such as artist name, song name etc. A few methods used semantic feature like music style. Tags and other text information are used in music retrieval. Content-based Music Information Retrieval (CBMR) is gaining widespread attention and could be helpful, since it forsakes the need of keyword.

However, there is few research on retrieval model on query-by-semantic description or other CBMR fusion method. Pitch or tempo, or other low-level feature is used in CBMR, these features cannot be transformed to high level semantic features directly. Relative research [2] shows that there may have a "semantic gap", which means the lack of association between low level physical features and semantic concept.

To solve the semantic gap problem, many recent researches on MIR is focus on music content and semantic expressions. Jun [3] proposed an ontology where the low-level and high-level descriptors collaborate to support semantics-based MIR. Buccoli [4] propose a Dimensional Contextual Semantic Model for defining semantic relations among descriptors in a context-aware fashion. This model is used for developing a

J. Wen et al. (Eds.): CCIR 2017, LNCS 10390, pp. 237–248, 2017.
https://doi.org/10.1007/978-3-319-68699-8_19

semantic music search engine [5]. Miotto and Lanckriet [6] proposed a Dirichlet mixture model (DMM) to improve automatic music annotation. In [7], they proposed a new approach that retrieves music using fuzzy music-sense features and audio features, which is a new method on semantic and CBMR. Foster [8] proposed string compressibility as a descriptor of temporal structure in audio, for determining musical similarity. The descriptors are based on computing track wise compression rates of quantized audio features, using multiple temporal resolutions and quantization granularities. In those related work, Turnbull [9–11] presented a concept of Query-by-Semantic-Description (QBSD), it is a natural way to retrieval in large music database, to overcome the lack of a cleanly-labeled, publicly-available, heterogeneous data set of songs and associated annotations, they collected CAL500 data set by having humans listen to and annotate songs using a survey designed to capture semantic associations between music and words. The methods adapted the supervised multi-class labeling (SML) model, used the CAL500 data to learn a model, annotated a novel song with meaningful words and retrieve relevant songs given a multi-word, text-based query.

The study shows that, music retrieval focusses on narrow the gap between music content and music semantic meaning. We present a query by example, a semantic vector is extracted by example song and used as query, searched in auto-annotated music database, the output of the retrieval system is a song list with the most similar songs in semantic vector space. Unlike other QBSD system, our system has two distinct advantages: firstly, instead of query by text, query by example song is a more convenient way for MIR. On the other hand, the similarity in semantic space which means more semantic same tags, is more exact than several words. The core problem lies on the auto-annotation efficiency by SML. If the model could not annotate the songs exactly, the retrieval result cannot satisfy the needs. Thus, the motivation of this paper is to improve effectiveness and the system feasibility. In consideration of the current study in deep learning methods on audio, we proposed a SML based on Convolutional Neural Network and improve the annotation algorithm by SMOTE. Experiments show the performance of this model is well, which provide a novel method for the QBSD system.

2 Related Work

2.1 The Architecture of the QBSD System

The architecture of the QBSD system is shown in Fig. 1. First, low-level feature, such as MFCCs, is used as the input of a well-studied deep learning network, then CNN map the audio signal into semantic space, by annotated the songs with semantic tags. Meanwell, music in database is also auto-annotated by the same mode. At last, query example and the songs in database are matched in similarity of semantic space, the most similar songs are the feedback as the retrieval or recommend results. By this system, users could receive better experience in a natural way, and get a result they really want.

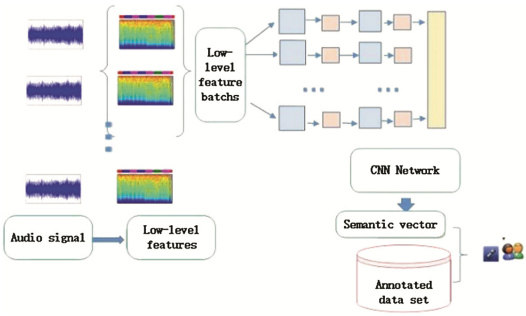

Fig. 1. The architecture of query by semantic description system

2.2 The Deep Learning Algorithms in Audio Signal Process

In recent years, deep learning method is widely employed in research image classification, image semantic mapping, audio classification and recognition, and get excellent efficiency. Deep learning method on audio process always use convolutional deep belief network to extract features for classification and retrieval. In [12], Deep Belief Network (DBN) was used to feature extraction, the learned features perform significantly better than MFCCs and obtained a classification accuracy of 84.3% on the Tzanetakis dataset. Dieleman [13] built a convolutional network that is then trained to perform artist recognition, genre recognition and key detection, improved accuracy for the genre recognition and artist recognition tasks. Hu Zhen [14] presented a hybrid model based on deep belief network (DBN) and stacked denoising autoencoder (SDA) to identify the composer from audio signal, the model got an accuracy of 76.26% in testing data set. Humphrey [15] learned a robust Tonnetz-space transform for automatic chord recognition. Hamel [16] proposed a feature extraction system consists of a Deep Belief Network (DBN) on Discrete Fourier Transforms (DFTs) of the audio, and the learned features perform significantly better than MFCCs. In a word, hidden Markov models (HMMs) to deal with the temporal variability of speech and Gaussian mixture models (GMMs) traditionally, however, as Hinton [17] said, Deep neural networks (DNNs) that have many hidden layers and are trained using new methods have been shown to outperform GMMs on a variety of speech recognition benchmarks, sometimes by a large margin. But deep learning algorithms should be learned by enough data and the structure of network needed finetuning. Therefore, a suitable network should be designed for the specific problem requirement.

To compare with GMM, we experiment on the CAL500 data set [11], which is composed by 500 songs of 500 unique artists, each annotated by a minimum of 3 individuals using a 174-tag vocabulary representing genres, instruments, emotions and other musically relevant concepts. The number of songs is minor. On the other hand, the number of frames in each music piece is huge (10000 frames in a song). Consider by the data characteristics, we employ the synthetic minority oversampling technique(SMOTE) [18], which is an over-sample method and the core idea was to construct the synthetic minority samples through the minority training data and its k nearest

neighborhoods, to overcome the shortage of the data set and improve the annotation efficiency.

3 QBSD Based on CNN

3.1 Problem Statement

Assume that a labeled training music dataset $D \equiv \{(x_i, y_j)\}_{j=1}^{c}$ is given, where each music is represented by a d-dimensional feature vector x_i, $x_i \in X$, X is the data set, i is the number of music pieces, y_j is distinct semantic labels available for training, j is the number of labels. The semantic vectors $S \equiv \langle y_0, y_1, \ldots, y_c \rangle$ is learned, c is the number of labels in semantic vectors. QBSD by example is addressed by learning a mapping from input features x_{input} to semantic label vectors S_x using a CNN model on the human annotation data set D, and the output of the system is a song list X_{list}, according to similarity between semantic label vectors S_x in data set and the input.

In the supervised multi-class labeling (SML) model [9], the probability distribution of each label in the semantic space was calculated by Gauss mixture model (GMM). The drawback of the nonparametric estimation technique is that the number of mixture components in the word-level distribution grows with the size of the training database. In practice, it may have to evaluate thousands of multivariate Gaussian distributions for each feature vectors of a novel query track. The training data should be subsampled or used mixture hierarchies' estimation, but they are not efficient in annotation or time cost.

3.2 Model Description

Audio signal has short-time stationarity and periodic features in long-time. The audio signals are segmented by a window, and lower-level features, include zero-crossings, centroid, rolloff and MFCC etc., are extracted traditionally. While, self-similarity is a common property in music pieces, melody may be repetitive in a song. CNN could be used to learn features in local receptive fields, thus music piece is the process unit in our algorithm and try to find new features for SML model.

39-dimensional MFCCs feature is extracted from the segmented audio signal frame, and five frames are cascaded as a long-frame that has 195-dimensional MFCCs. Then fifty long-frames is treated as a music piece, which is a two-dimension vector with a size of 195×10. The music piece is the input of a CNN, and the output of the net is:

$$h_{ij}^{k} = \theta((W^{k} * x)_{ij} + b_k) \tag{1}$$

Where k is the k-th filter, x is the input music piece feature batch, W^k denote the parameter (or weight) associated with filter k. b_k is the bias associated with filter k. (i, j) is the location on feature batch. Convolutional features are calculated after the activation function θ, which usually uses sigmoid or tanh function. We choose relu, because it results in the neural network training several times faster [19], without making a

significant difference to generalization accuracy. The architecture based on CNN is shown in Fig. 2.

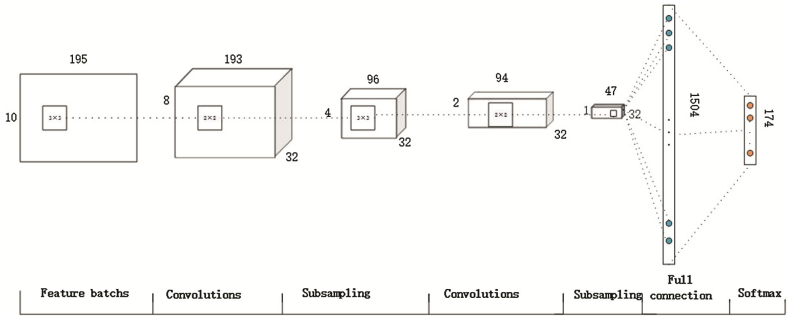

Fig. 2. The architecture of CNN

There are two convolutional layers, two pooling layers, one fully connected layer and a loss layer in this network, the size of each layer is shown in Fig. 2. We choose 32 convolutional filters in one layer, the shape of the filter is 3×3 and the max pooling shape is 2×2. $x_{flatten}$ donate the output of the fully connected layer, then the output of the last lay y_{out} is shown as follow:

$$y_{out} = softmax\left(x_{flatten}\right) = \frac{1}{1 + e^{-x_{flatten}}} \tag{2}$$

The value of y_{out} is in $(0, 1)$, the network training penalizes the deviation between y_{out} and the human annotated semantic label y_j, c the number of labels:

$$E^N = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{c} \left(y_j^{nk} - y_{out}^{nk} \right)^2 \tag{3}$$

E^N is the total loss of the N training songs, y_j^{nk} denote human annotated label of the n th sample on the kth label, y_{out}^{nk} is the output of the n th sample on the kth label, then the network weights are fine-tuned according to the loss.

Loss function adjustment. Consider imbalanced distribution as prior information. For each song S in data set, S is annotated by N labels, according to CNN structure, the output from last level softmax might be $\{\ldots, 0 \ldots, \frac{1}{N}, \ldots, \frac{1}{N}, \ldots, 0 \ldots\}$. We adjust the output by adding a weight w_i, which in inverse proportion of annotation frequency. Let f_i be the number of annotation samples for label i, the smaller f_i is, and the bigger w_i is. The output of the net turns into $\{\ldots, 0 \ldots, w_i \frac{1}{N}, \ldots, w_j \frac{1}{N}, \ldots, 0 \ldots\}$, $\sum_{i=0}^{N} w_i \frac{1}{N} = 1$, then:

$$w_i = 1 - \frac{f_i}{\sum_{i=0}^{N} f_i} + \frac{1}{N} \tag{4}$$

The loss function is adjusted as:

$$E^N = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{c} w_i\left(y_i^{nk} - y_{out}^{nk}\right)^2 \tag{5}$$

Unlike a usual multi-classification problem, music piece can be annotated by many labels, it means that a song can belong be more than one class according to the label. In most multi-classification, samples are balanced (the number of samples is almost equal) in each class. However, music annotation is an unbalanced, in the human annotated dataset, one label may be annotated on many songs, or on the other side, only a few songs is annotated by it. If the difference of the annotation frequency is ignored, the learned model could not give the correct label for a song, for example, 'happy' may be annotated on 99% of the songs in data set, the learned classifier assigns all songs with the label happy would still achieve the accuracy of 99%. However, due to the low recall ratio for the minority, such extreme result is not what we have desired.

To solve imbalance learning problem, we employ two strategies for the learning model. First, the adjustment of the weights of errors in the loss function, which is a direct way to reduce the impact of imbalance. Another method is synthetic minority oversampling technique(SMOTE) proposed by Chawla etc. [18]. CAL500 is a small data set, human annotation is costly work, we conduct the synthetic minority samples for training data, thus the model would learn more information from minority labels.

SMOTE algorithm. SMOTE is a typical oversampling method with universal applications and the concrete process for generating the synthetic samples can be described as shown in Table 1:

Table 1. SMOTE algorithm

Algorithm 1: SMOTE ALGORITHM
Input: Each sample in data set x_i;
Output: New synthetic sample for the minority x_{new}; **Description:** 1.Calculate the K-nearest neighbors for each sample x_i, 2.Select a random number δ, generated from a uniform distribution U[0,1]; 3.Output a new synthetic sample for minority as: $\boldsymbol{x_{new} = x_i + \delta(x_i' - x_i)}$

The result of SMOTE, implemented by [20], is shown in Fig. 3. The number of samples increased and the data from each class becomes balanced.

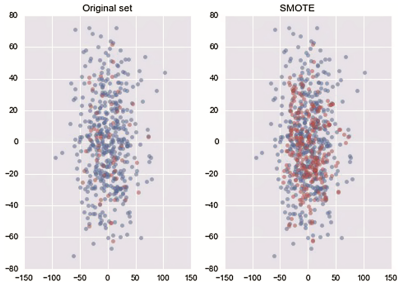

Fig. 3. Result of SMOTE algorithm

3.3 Retrieval Algorithm

Query by example algorithm is shown in Table 2:

Table 2. Retrieval algorithm

Algorithm 2: Retrieval algorithm with example

Input: A labeled training music dataset D ≡ $\{(x_i, y_j)\}_{j=1}^{c}$, each music is represented by a d-dimensional feature vector x_i, $x_i \in X$, X is the data set, i is the number of music pieces, j is the number of labels; A new query song q;
Output: The output of the system is a song list X_{list};
Description:

1. Training a CNN network shown in Fig.2 ,according to annotation relation between x_i,y_j,get the net parameters W^k, b_k , and the input y_{out}^p;
2. Calculate semantic vectors of each song x_i:

$$S_{sementic}^{X} = \frac{1}{p}\sum_{j=1}^{p} y_{out}^{p}$$

3. Calculate semantic vector of the query q, $S_{sementic}^{q}$;
4. Compute the Cosine similarity R_s between $S_{sementic}^{X}$ and $S_{sementic}^{q}$;
5. Let X_{list} be a list of top x candidate songs with the highest R_s;
6. Return X_{list}.

4 Experiments and Analysis

4.1 Data Set and Features

To evaluate the performance of the proposed approach, we use CAL500 data set [9], which has 500 songs by 500 unique artists each annotated by minimum of 3 individuals using a 174-tag vocabulary. A song is annotated with a tag if 80% of the human annotators agree that the tag is relevant, the value is 1 in semantic vector, otherwise the value is 0. In our experiments, 39-dimensional MFCCs are used as CNN input, 174-dimensional semantic vector as retrieval model output. CNN model is

implemented by Keras [21], which is a high-level neural networks API, written in Python and capable of running on top of either TensorFlow or Theano. Hyper-parameters for CNN is shown in Table 3. For batch processing, we align each song with 10000 frames, five frames are cascaded as a long-frame that has 195-dimensional MFCCs. Then 50 long-frames is treated as a music piece, which is a two-dimension vector with a size of 195×10, 200 music pieces in each song.

Table 3. Hyper-parameters for CNN

Hyper-parameter	Value
size of batch	4
number of classes	174
number of training epoch	30
number of convolutional filters to use	32
size of pooling area for max pooling	2
convolution kernel size	3×3
optimizer	SGD
learning rate	0.1
decay	1e-6
momentum	0.9
nesterov	True

4.2 Evaluation of Annotation and Retrieval

Annotation performance is measured following the procedure described by Coviello etc. [22]. Annotation accuracy is reported by computing precision, recall and F-score for each tag and then averaging over all tags. Per-tag precision is the probability that the model correctly uses the tag when annotating a song. Per-tag recall is the probability that the model annotates a song that should have been annotated with the tag. Precision, recall and F-score measure for a tag are defined as:

$$P = |W_C| / |W_A|, R = |W_C| / |W_H|,$$

$$F = 2((P)^{-1} + (R)^{-1})^{-1} \tag{6}$$

Where $|W_H|$ is the number of tracks that have W in the ground truth, $|W_A|$ is the number of times our annotation system uses when W automatically tagging a song, and $|W_C|$ is the number of times is W correctly used.

To evaluate retrieval performance, we report mean average precision (MAP), area under the receiver operating characteristic curve (AROC) and averaged over all the query tags. The ROC curve is a plot of true positive rate versus false positive rate as we

move down the ranked list. The AROC is obtained by integrating the ROC curve, and it is upper bounded by 1. Random guessing would result in an AROC of 0.5.

4.3 Result Analysis

Annotation and retrieval on tags. First, we search each tag in the data set, list is ranked by decision value, and results are averaged on tags. The results are shown in Table 4, CNN-AD (loss function adjustment on CNN), CMM-SMOTE (SMOTE on train data) are better than SVM and HEM-GMM on annotation precision and AROC.

Table 4. Annotation and retrieval results for various algorithms on the CAL500 data set

	Annotation			Retrieval	
Model	P	R	F-score	AROC	MAP
HEM-GMM	**0.49**	0.33	0.26	0.66	0.45
SVM	0.46	0.46	0.44	**0.72**	0.50
CNN-AD	**0.49**	0.46	**0.45**	0.71	**0.52**
CNN-SMOTE	0.49	0.47	0.46	**0.72**	**0.52**

Retrieval by tag 'ACOUSTIC GUITAR' results is shown in Table 4, CNN-AD and CNN-SMOTE are better than CNN-SVM.

Retrieval by query example. Query based on example is used the whole song as a query, the example is annotated by the learned model and transformed into a semantic vector, then we calculate the similarity between the example semantic vector and the human-annotated data set, a song list is ranked by Cosine similarity. In our experiment, if the annotation accuracy is well, the same human-annotated song would be the output. Thus, we define Hit@k as the percentage that the same song output from retrieval, could be used to compare different models. The results are shown in Fig. 4. CNN-SMOTE and CNN-AD are better than CNN-SVM on Hit@10, and Hit@10 values are more than 92%, which means query by example is better than several text descriptions and nearby human-annotation (Table 5).

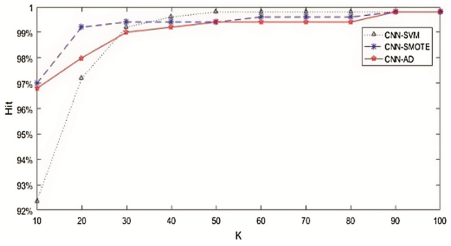

Fig. 4. Hit@k performance of our model on CAL500

Table 5. Top-10 retrieved songs for "ACOUSTIC GUITAR." Songs with acoustic guitar are marked in bold

Rank	CNN-SVM	
	Artist	Song Name
1	**Myles cochran**	**Getting stronger**
2	**Van morrison**	**And It Stoned Me**
3	**Buena Vista Social Club**	**El Cuarto de Tula**
4	**The Black Crowes**	**Thorn in My Pride e**
5	**Johnny Cash**	**The Man Comes Around**
6	**George Harrison**	**All Things Must Pass**
7	R.E.M	Camera
8	The Monkees	A Little Bit Me a Little Bit You
9	LOVE	You Set the Scene
10	Wicked Allstars	Happy
Rank	CNN-AD	
	Artist	Song Name
1	**Black Crowes**	**Thorn in My Pride**
2	**Gram parsons**	**$ 1000 wedding**
3	**Neil Young**	**Razor Love**
4	**Bob Dylan**	**I'll Be Your Baby To-night**
5	Mr Gelatine	Knysnamushrooms
6	**New Order**	**Blue Monday**
7	**Myles Cochran**	**Getting stronger**
8	The Rolling Stones	Little by Little
9	They Might Be Giants	I Should Be Allowed to Think
10	Ultravox	Dancing with Tears in My Eyes
Rank	CNN-SMOTE	
	Artist	Song Name
1	**Myles Cochran**	**Getting stronger**
2	**Van Morrison**	**And It Stoned Me**
3	**Buena Vista Social Club**	**El Cuarto De Tula**
4	**Black Crowes**	**Thorn in My Pride**
5	**Johnny Cash**	**The Man Comes Around**
6	**George Harrison**	**All Things Must Pass**
7	Brenton Wood	Lovey Dovey Kind of Love
8	Dirt	THE STOOGES
9	Buddy Holly	Peggy Sue
10	**Air**	**Sexy Boy**

5 Conclusions

In this paper, we proposed a music retrieval model based on query example semantic description, CNN is used to learn a Supervised Multi-class labeling system, by which example query is transformed to a semantic vector, and searched in the data set. To improve the annotation accuracy, loss function adjustment and SMOTE algorithm are employed, the results show that an example song instead of only a few texts, could get a result more semantically similarity and it is a more natural way to find what we want. Loss function adjustment method based on CNN, the model should adjust the weight by labeling frequency, low frequency tags would not be ignored in learning process. SMOTE algorithm produces more samples for low frequency tags and get better annotation result, but it should be used on each tag in vocabulary, which means we should learn different models for each tag and costs much more time. In future work, we will design and test more different networks for semantic tags annotation and large-scale music data set unsupervised multi-class annotation algorithm should be considered.

Acknowledgment. Supported by the National Natural Science Foundation of China (Grant No. 61632011); the National Natural Science Foundation of China (Grant No. 61562080); the National Natural Science Foundation of China (Grant No. 61602079)

References

1. BigData-Research. http://www.bigdata-research.cn/content/201606/285. 12 June 2016
2. Casey, M., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., Slaney, M.: Content-based music information retrieval: current directions and future challenges. Proc. IEEE **96**(4), 668–696 (2008)
3. Wang, J., Deng, H., Yan, Q.: A collaborative model of low-level and high-level descriptors for semantics-based music information retrieval. In: International Conference on Web Intelligence and Intelligent Agent Technology, pp. 532–535. IEEE, New York (2008)
4. Buccoli, M., Gallo, A., Zanoni, M., Sarti, A., Tubaro, S.: A dimensional contextual semantic model for music description and retrieval. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 673–677. IEEE, New York (2015)
5. Buccoli, M., Zanoni, M., Sarti, A., Tubaro, S.: A music search engine based on semantic text-based query. In: IEEE International Workshop on Multimedia Signal Processing, pp. 254–259. IEEE, New York (2013)
6. Miotto, R., Lanckriet, G.: A generative context model for semantic music annotation and retrieval. IEEE Trans. Audio Speech Lang. Process. **20**(4), 1096–1108 (2012)
7. Su, J.H., Wang, C.Y., Chiu, T.W., Ying, J.C., Tseng, V.S.: Semantic content-based music retrieval using audio and fuzzy-music-sense features. In: IEEE International Conference on Granular Computing, pp. 259–264. IEEE, New York (2014)
8. Foster, P., Mauch, M., Dixon, S.: Sequential complexity as a descriptor for musical similarity. IEEE Press **22**(12), 1965–1977 (2014)
9. Turnbull, D., Barrington, L., Torres, D., Lanckriet, G.: Towards musical query- by- semantic description using the CAL500 data set. In: International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 439–446. ACM, New York (2007)

10. Turnbull, D., Barrington, L., Torres, D., Lanckriet, G.: Semantic annotation and retrieval of music and sound effects. IEEE Trans. Audio Speech Lang. Process. **16**(2), 467–476 (2008)
11. Turnbull, D.R., Barrington, L., Lanckriet, G., Yazdani, M.: Combining audio content and social context for semantic music discovery. In: International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 387–394. ACM, New York (2009)
12. Lee, H., Yan, L., Pham, P., Ng, A.Y.: Unsupervised feature learning for audio classification using convolutional deep belief networks. In: International Conference on Neural Information Processing Systems, pp. 1096–1104. Springer, Heidelberg (2009)
13. Dieleman, S., Brakel, P., Schrauwen, B.: Audio-based music classification with a pretrained convolutional network. In: Proceedings of the ISMIR (2011)
14. Hu, Z., Fu, K., Zhang, C.: Audio classical composer identification by deep neural network. J. Comput. Res. Dev. **51**(9), 1945–1954 (2014)
15. Humphrey, E.J., Cho, T., Bello, J.P.: Learning a robust Tonnetz-space transform for automatic chord recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 453–456. IEEE, New York (2012)
16. Hamel, P., Eck, D.: Learning features from music audio with deep belief networks. In: Proceedings of the ISMIR, pp. 339–344 (2010)
17. Hinton, G., Deng, L., Yu, D., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Sig. Process. Mag. **29**(6), 82–97 (2012)
18. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**(1), 321–357 (2002)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105. ACM, New York (2012)
20. Lemaitre, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res. **18**(17), 1–5 (2017)
21. Chollet, F.: Keras, GitHub repository (2015). https://github.com/fchollet/keras
22. Coviello, E., Chan, A.B., Lanckriet, G.: Time series models for semantic music annotation. IEEE Trans. Audio Speech Lang. Process. **19**(5), 1343–1359 (2011)