Deep Cross-Modal Correlation
Audio and Lyrics in Music Retr

Deep Cross-Modal Correlation Learning for Audio and Lyrics in Music Retrieval

YI YU, National Institute of Informatics, Japan
SUHUA TANG, The University of Electro-Communications, Japan
FRANCISCO RAPOSO*, Universidade de Lisboa, Portugal
LEI CHEN, Hong Kong University of Science and Technology

Deep cross-modal learning has successfully demonstrated excellent performance in cross-modal multimedia retrieval, with the aim of learning joint representations between different data modalities. Unfortunately, little research focuses on cross-modal correlation learning where temporal structures of different data modalities such as audio and lyrics should be taken into account. Stemming from the characteristic of temporal structures of music in nature, we are motivated to learn the deep sequential correlation between audio and lyrics. In this work, we propose a deep cross-modal correlation learning architecture involving two-branch deep neural networks for audio modality and text modality (lyrics). Data in different modalities are converted to the same canonical space where inter modal canonical correlation analysis is utilized as an objective function to calculate the similarity of temporal structures. This is the first study that uses deep architectures for learning the temporal correlation between audio and lyrics. A pre-trained Doc2Vec model followed by fully-connected layers is used to represent lyrics. Two significant contributions are made in the audio branch, as follows: i) We propose an end-to-end network to learn cross-modal correlation between audio and lyrics, where feature extraction and correlation learning are simultaneously performed and joint representation is learned by considering temporal structures. ii) As for feature extraction, we further represent an audio signal by a short sequence of local summaries (VGG16 features) and apply a recurrent neural network to compute a compact feature that better learns temporal structures of music audio. Experimental results, using audio to retrieve lyrics or using lyrics to retrieve audio, verify the effectiveness of the proposed deep correlation learning architectures in cross-modal music retrieval.

CCS Concepts: • **Information systems** → **Music retrieval**; Information extraction;

Additional Key Words and Phrases: Convolutional neural networks, Deep cross-modal models, Correlation learning between audio and lyrics, Cross-modal music retrieval, Music knowledge discovery

ACM Reference Format:
Yi Yu, Suhua Tang, Francisco Raposo, and Lei Chen. 2010. Deep Cross-Modal Correlation Learning for Audio and Lyrics in Music Retrieval. *ACM Trans. Multimedia Comput. Commun. Appl.* 9, 4, Article 39 (March 2010), 16 pages. https://doi.org/0000001.0000001

*Francisco was involved in this work during his internship in National Institute of Informatics (NII), Tokyo.

Authors' addresses: Yi Yu, National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan; Suhua Tang, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo, 182-8585, Japan; Francisco Raposo, Universidade de Lisboa, INESC-ID Lisboa R. Alves Redol 9, Lisboa, 1000-029, Portugal; Lei Chen, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.

1551-6857/2010/3-ART39 $15.00
https://doi.org/0000001.0000001

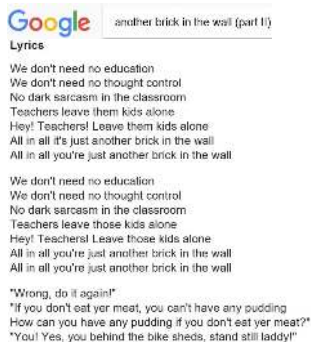

Fig. 1. Google lyrics for song title "another brick in the wall (part II)".

1 INTRODUCTION

Music audio and lyrics provide complementary information in understanding the richness of human beings' cultures and activities [27]. Music[1] is an art expression whose medium is sound organized in time, and previous work has investigated content-based music retrieval [35, 36, 41]. Lyrics[2] as natural language represent music theme and story, which are a very important element for creating a meaningful impression of the music. Starting from the late 2014, Google provides music search results containing song lyrics when given a specific song title, as shown in Fig. 1. Sometimes, however, users may also like to search lyrics with a recorded audio sample, or vice versa, and music composers may want to find similar lyrics or audio tracks when they create new songs (for reference or for avoiding duplication), which requires cross-modal music retrieval. In particular, let us imagine a typical scenario where a user sits in a Starbucks Coffee and suddenly a song attracts her attention. She records it by her iphone and would like to immediately find the lyrics with the recorded audio sample. One may choose to realize the cross-modal retrieval by two unimodal retrievals, e.g., find song title from audio content and then find the lyrics, or vice versa. This, however, requires (i) a database contain all lyrics, (ii) a database contain all audio tracks, and (iii) meta data (such as song title and artist name) that associate lyrics with audio tracks. The majority of audio tracks are created with metadata such as song title and artist. Recently, online services, such as Shazam and SoundHound, predict song title with an audio slice (recorded with environmental noise). It seems possible to realize the task of lyrics query with audio by two unimodal retrievals, although it fails when either modality corresponding to a query is not included in the database. Unfortunately, many songs, especially old ones, do not come with lyrics in the electronic form, and the pairing relationship between audio tracks and lyrics is not always available. Therefore, it is better to directly learn the cross-modal correlation between lyrics and audios.

Searching lyrics by audio was almost impossible years ago due to the limited availability of large volumes of music audio and lyrics. The profusion of online music audio and lyrics from music sharing websites, such as Spotify, YouTube, MetroLyrics, Azlyrics, and Genius, shows the opportunity to understand musical knowledge from content-based audio and lyrics, by leveraging large volumes of cross-modal music data aggregated on the Internet.

Motivated by the fact that audio content and lyrics are very fundamental aspects for understanding what kind of cultures and activities a song wants to convey, this research pays attention to deep correlation learning between audio and lyrics for cross-modal music retrieval and considers

[1]https://en.wikipedia.org/wiki/Music
[2]https://en.wikipedia.org/wiki/Lyrics

two real-world tasks: using audio to retrieve lyrics or vice versa. Several contributions are made in this paper, as follows:

i) This work studies cross-modal music retrieval by using either audio or lyrics as a query to find its counterpart in the other modality. To the best of our knowledge, this is the first work that leverages a deep architecture to learn the correlation between audio tracks and lyrics.

ii) Data in different modalities are projected to the shared space where inter modal canonical correlation analysis is exploited as an objective function to calculate the similarity of temporal structures. Deep neural networks (DNNs) such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) are used to learn audio representation, and they are optimized in the end-to-end network structure for the correlation analysis between audio and lyrics.

iii) Extensive experiments confirm the effectiveness of our deep correlation learning architecture for audio-lyrics music retrieval, which are meaningful results and studies for attracting more efforts on mining music knowledge structure and correlation between data in different modalities.

The rest of this paper is structured as follows. Research motivation and background are introduced in Sec.2. Then, Sec.3 presents why and how we build a deep correlation learning architecture for audio-lyrics music retrieval. Experimental evaluation results are shown in Sec.4. Finally, conclusions are pointed out in Sec.5.

2 MOTIVATION AND BACKGROUND

Music has permeated our daily life, which contains different modalities in real-world scenarios such as temporal audio signal, lyrics with meaningful sentences, high-level semantic tags, and temporal visual content. The widespread availability of large-scale multimodal music data brings us research opportunity to tackle cross-modal music retrieval. In the following, we review related techniques, including CNN, music classification by audio/lyrics, cross-modal retrieval between music and image/video, deep cross-modal learning between image and text.

2.1 Convolutional Neural Networks (CNNs)

CNNs have been successfully exploited to handle various tasks in the field of computer vision and multimedia [16, 31]. Different kernels (filters) are used in CNNs to capture different local patterns, and this will generate multiple intermediate feature maps (called channels). Specifically, the convolutional operation in one convolutional layer is defined as

$$\boldsymbol{x}^j = f(\sum_{k=0}^{K-1} \boldsymbol{H}^{jk} \otimes \boldsymbol{s}^k + a^j), \tag{1}$$

where the superscripts j, k are channel indices, $\boldsymbol{s}^k$ is the k-th input channel, $\boldsymbol{x}^j$ is the j-th output channel, $\otimes$ is the convolutional operation, $\boldsymbol{H}^{jk}$ is the convolutional kernel (or the filter) that associates the k-th input channel with the j-th output channel, a^j is the bias for the j-th channel, and $f(\cdot)$ is a non-linear activation function. All weights that define a convolutional layer are represented as a 4-dimensional array with a shape of (h, l, K, J), where h and l determine the kernel size, and K and J are the number of input and output channels, respectively.

A 2-D convolutional kernel $\boldsymbol{H}^{jk}$, as a common filter, is applied to the whole input channel. This kernel is shifted along both axes and a local correlation is computed between the kernel and input. The kernels are trained to find local salient patterns that maximize the overall objective. As a kernel sweeps the input, it generates a new output in order, which preserves the spatiality of the input. Convolutional layers are often followed by pooling layers, which reduce the size of feature map by down sampling them. The max function is a typical pooling operation. This selects the maximal value from a pooling region, instead of keeping all information in the region.

CNN is also applied to the tasks of music information retrieval such as genre classification [6], acoustic event detection [10], automatic music tagging [5]. In such cases, the spectrogram of audio signal is usually regarded as an image. When lacking computational power and large annotated datasets, it is preferred to directly use pre-trained CNNs such as VGG16 [31] to extract features [10], or further combine it with fully-connected layers to extract semantic features.

Besides using the 2-D CNN in the similar way as in image processing, it is possible to directly apply 1-D strided convolution on the waveform of audio signal [19], which incorporates the computation of spectral feature into the filters. The stride length and filter length usually are set large enough to capture short-term spectral features of audio signals. It is also possible to apply 1-D convolution on the spectrogram or MFCC sequence to learn temporal representations [18].

CNN is a key component of this work, and we focus on 2-D CNN.

2.2 Lyrics and Audio in Music Classification

Recent research has shown how to use lyrics, audio, or their combination, in semantic audio classification such as emotion or genre recognition in music.

For example, authors in [24] proposed an unsupervised learning method for mood recognition where Canonical Correlation Analysis (CCA) was applied to identify correlations between lyrics and audio, and the evaluation of mood classification was done based on the valence-arousal space. An interesting corpus with each song in the MIDI format and emotion annotation is introduced in [25]. Coarse-grained classification for six emotions is learned by support vector machines (SVM), and this work showed that either textual feature or audio feature can be used for emotion classification, and their joint use leads to a significant improvement. Emotion lyrics datasets in English [21] are annotated with continuous arousal and valence values. Specific text emotion attributes are considered to complement music emotion recognition. Experiments on the regression and classification of music lyrics by quadrant, arousal, and valence categories are performed. Application of hierarchical attention network is proposed in [32] to handle genre classification of intact lyrics. This network is able to pay attention to words, lines, and segments of the song lyrics, where the importance of words, lines, and segments in layer structure is learned.

Distinct from previous research on music classification by using lyrics and audio, our work focuses on audio-lyrics cross-modal music retrieval: using audio to retrieve lyrics or vice versa. This is a very natural way for us to retrieve lyrics or audio on the Internet. However, no much research has investigated this task.

2.3 Cross-modal Music/Image(Video) Retrieval

Some existing researches on cross-modal music retrieval intensively focus on investigating music and visual modalities [1, 3, 7, 23, 26, 29, 33, 38].

A model, capturing the similarity between audio features extracted from music song and visual features extracted from the album covers, is trained by a Java SOMToolbox framework in [23]. Then, according to this similarity, people can organize a music collection and make use of album cover as visual content to retrieve a song from multimodal music data. Based on multimodal mixture models, a statistical method to jointly modeling music, images, and text [3] is used to support retrieval over a multimodal dataset. Lyrics-based music attributes are utilized for image representation in [33]. Cross-modal ranking analysis is suggested to learn semantic similarity between music and image, with the aim of obtaining the optimal embedding spaces for music and image.

To generate a soundtrack for an outdoor video, an effective heuristic ranking method is suggested based on heterogeneous late fusion by jointly considering venue categories, visual scene, and user listening history [29]. Confidence scores, produced by SVM^{hmm} models constructed from geographic, visual, and audio features, are combined to obtain different types of video characteristics.

To learn the semantic correlation between music and video, a novel approach to selecting features and statistical novelty based on kernel methods [7] is proposed for music segmentation. Co-occurring changes in audio and video content of music videos can be detected, where the correlations can be used in cross-modal audio-visual music retrieval. A content-based music video retrieval method using soft intra-modal structure constraint is studied in [12], which leverages the relative distance relationship between intra-modal samples before embedding.

Distinct from intensive research that use metadata of different music modalities in cross-modal music retrieval, our work focuses on a deep architecture for content-based cross-modal music retrieval, based on correlation learning between audio and lyrics.

2.4 Deep Cross-modal Learning between Image and Text

We have witnessed several efforts devoted to investigating cross-modal learning between different modalities, such as [4, 14, 15, 37, 39, 42], to facilitate cross-modal matching and retrieval. Latest studies extensively pay attention to deep cross-modal learning between image and textual descriptions such as [15, 34, 37, 39]. Most existing deep models with two-branch sub-networks explore pre-trained CNN [31] as image branch [39] and utilize pre-trained document-level embedding model [17] or hand-crafted feature extraction such as bag of words [15] as text branch. Image and text modalities are converted to the joint embedding space to calculate a single ranking loss function by a feed-forward way. Image-text benchmarks such as [20, 28, 40] are used to evaluate the performances of cross-modal matching and retrieval.

Existing deep cross-modal retrieval methods have two properties: i) Cross-modal correlation between image and text is learned without considering temporal sequences. ii) Pre-trained models are directly applied to represent image or text. Distinct from existing deep cross-modal retrieval architectures, this work takes into account temporal sequences to learn the correlation between audio and lyrics for facilitating audio-lyrics cross-modal music retrieval, where sequential audio and lyrics are converted to the canonical space and a neural network with two-branch sequential structures for audio and lyrics is trained.

3 DEEP AUDIO-LYRICS CORRELATION LEARNING

We develop a deep cross-modal correlation learning architecture that predicts latent alignment between audio and lyrics, which enables audio-to-lyrics or lyrics-to-audio music retrieval. In this section, we explain how our deep architecture is learned. Specifically, we investigate different deep network models for correlation analysis and different deep learning methods for audio feature extraction.

Figure 2 shows the proposed end-to-end deep DCCA network, which aims at simultaneously learning the feature extraction network (CNN or RNN for audio branch) and the non-linear embedding network for correlation analysis between audio and lyrics. This model is degenerated to a simple DCCA network, when the CNN/RNN model is replaced by a pre-trained model that is only used for feature extraction but without re-training. Then, we study the pure effect of DCCA in the correlation analysis.

In the following, we explain deep audio feature extraction, deep textual feature extraction, non-linear embedding, and CCA analysis, respectively.

3.1 Deep Audio Feature

A music audio signal usually is represented as a 2-D spectrogram, which preserves both its spectral and temporal properties. However, it is difficult to directly use this for the DCCA analysis, due to its high dimension. Therefore, we investigate different methods for the dimension reduction.

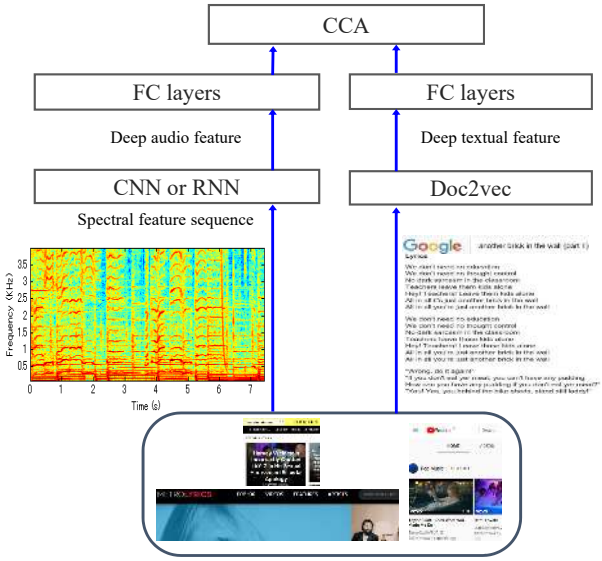

Fig. 2. Deep correlation learning between audio and lyrics.

3.1.1 Audio Feature from MFCC by CNN. MFCC is a very efficient feature for semantic genre classification [30] and music audio similarity comparison [8]. For each audio signal 30s long, it is resampled to 22,050Hz with a single channel. With frame length 2048 and step length 1024, a sequence of MFCCs (20x646, feature dimension is 20 per frame and there are 646 features) are computed. The MFCC sequence is further decimated to 4 sub sequences, each with 161 frames covering the whole length and associated with the same lyrics. These choices are made in order to consider both the kernel and the batch size in CNN-based deep learning. (1) As for the former, the same note (spectrum) in music audio signal typically lasts for a period, which may cover several adjacent frames. Such redundancy affects the detection of salient features by a small kernel (3x3 convolution). In contrast, using decimated MFCC sequence leads to more compact MFCC sequence with more variations in adjacent frames, which facilitates the salient feature detection. (2) As for the latter, when the system memory is fixed, compact MFCC sequences enable a larger batch size, which helps to calculate more stable statistical information for the learning.

To compute a single feature vector for correlation analysis, we successively apply convolutional layers with different kernels to capture local salient features, and use pooling layers to reduce the dimension. By inserting the pooling layer between adjacent convolutional layers, a kernel in the late layer corresponds to a larger kernel in the previous layer, and has more capacity in representing semantic information. Then, using small kernels in different convolutional layers can achieve the function of a large kernel in one convolutional layer, but is more robust to scale variance. In this sense, a combination of successive convolutional layers and pooling layers can capture features at different scales, and the kernels can learn to represent complex patterns.

For implementing an end-to-end deep learning, the configuration of CNN used for audio branch in this work is shown in Table 1. It consists of 3 convolutional layers and 3 max pooling layers, and outputs a feature vector with a size of 1536. We tried to add more convolutional layers but see no significant difference. Rectified linear unit (ReLU) is used as an activation function in each convolutional layer except the last one. Batch normalization is used before activation. Convolutional kernels (3x3) are used in every convolutional layer. These kernels help to learn local spectral-tempo

Table 1. Configuration of CNNs for audio branch with MFCC

MFCC: 20x646/4
Convolution, 3x3x48
Max-pooling (2,2), output 10x80x48
Convolution: 3x3x96
Max-pooling (3,3), output 3x26x96
Convolution: 3x3x192
Max-pooling (3,3), output 1536

structures. In this way, CNN converts an audio feature sequence (a 2-D matrix) to a high dimensional vector.

With the input spectrogram $\boldsymbol{s}$, the feature output by the convolutional layers is $\boldsymbol{x} = f_3(\boldsymbol{H}_3 \otimes f_2(\boldsymbol{H}_2 \otimes f_1(\boldsymbol{H}_1 \otimes \boldsymbol{s} + a_1) + a_2) + a_3)$, where $\boldsymbol{H}_i$, a_i and f_i are the convolutional kernel, bias, and activation function in the ith layer.

3.1.2 Audio Feature from Mel-spectrogram by CNN. We also use Mel-spectrogram (dimension is 96 per frame) together with CNN, which contains more detailed information. There are four convolutional layers, where each of the first three is followed by a max pooling layer, and the final output is 3072 dimension.

3.1.3 Audio Feature by Pre-trained CNN Model . We also investigate the pre-trained CNN model [5], which takes a log-amplitude Mel-spectrogram (96x1366) as input. It has 5 convolutional layers besides the output layer, and was originally trained on the Million Song Dataset, for classifying music songs into 50 tags. The 5 convolutional layers use the same kernel size (3x3) and the number of kernels are 64, 128, 128, 128, and 64, respectively, and the pooling settings after convolutional layers are (2,4), (2,4), (2,4), (3,5), (4,4), respectively. By using either average pooling or standard deviation pooling after each convolutional layer, a 32-dimension vector is generated per layer. Concatenating all of them together generates a feature vector of 320 dimension.

3.1.4 Audio Feature from Mel-spectrogram by RNN. Besides CNN, we also try to use RNN to extract compact features from the Mel-spectrogram so that it better captures temporal property. But in our experiments, we find that directly applying a long short term memory (LSTM) [11] model on a long sequence of audio features is difficult to achieve a good performance. Therefore, we first apply the pretrained VGG16 model [31] on the sequence of Mel-spectrogram to get local summaries, and use the shorter sequence of VGG16 features to represent an audio signal. To match the size of VGG16 (input dimension is 224x224), the Mel-spectrum per-frame is changed to 224. To get a good time resolution, frame length 1024 and step length 512 are used. In this way, each VGG16 feature spans about 5sec. With 50% overlap, a 30s-long audio generates a sequence of 12 VGG16 features. This VGG16 sequence will pass a LSTM model to extract a compact feature.

The VGG16 model was pre-trained on the ImageNet for classifying an image into one of 1000 classes. The VGG16 model consists of 13 convolutional layers (conv1-conv13) and three fully connected layers (fc14-fc16). All layers use a ReLU activation except fc16 which uses a softmax activation for the purpose of image classification. Each fully connected layer, except the last one, is followed by a dropout layer, to avoid overfitting. Images are processed sequentially per layer, and finally the 4,096-dimensional feature of fc15 is extracted as the visual feature for each venue image.

Table 2. Structure of sub-DNNs

	Sub-DNN1 (Audio)	Sub-DNN2 (Text)
1st layer	1024, sigmoid	1024, sigmoid
2nd layer	1024, sigmoid	1024, sigmoid
3rd layer (output)	D, linear	D, linear

3.1.5 Other Hand-crafted Feature. Spotify provides a hand-crafted 65-dimension feature for each audio track[3], with detailed information as follows: tempo (1), tempo confidence (2), time signature (3-7), time signature confidence (8), mode (9), mode confidence (10), number of sections (11), energy (12), danceability (13), mean Chroma pitches (14-25), standard deviation Chroma pitches (26-37), timbre mean (38-49), timbre standard deviations (50-61), loudness start mean (62), loudness start standard deviations (63), loudness max mean (64), loudness max standard deviations (65).

3.2 Deep Textual Feature

From the sequence of words in the lyrics, textual feature with a fixed length is computed, based on the concept of word embedding. Word embedding (Word2Vec) represents each word by a vector in a space where words with similar meaning are close to each other. Doc2Vec [17] extends the Word2Vec model by converting an entire document into a fixed length vector, taking into account the order of words in the context. When applying Doc2Vec, lyrics text of each song is tokenized by using coreNLP [22], and passed to the infer_vector module of the Doc2Vec model, generating a 300-dimensional feature for each song.

We use the pretrained apnews_dbow weights[4] in the experiment. They are trained on the Associated Press News dataset, for the Distributed Bag of Words (DBOW) model, which is used to represent a news article by a fixed length vector.

3.3 Non-linear Embedding

Audio features and textual features are further embedded into low dimensional features in a shared D-dimensional semantic space by using different sub DNNs composed of fully connected layers.

The details of sub DNNs are shown in Table 2. These two sub DNNs (each with 3 fully connected layers) implement the non-linear mapping of DCCA. The audio feature generated by the feature extraction part is denoted as $\boldsymbol{x} \in R^m$ (m varies with each method) and deep textual feature is denoted as $\boldsymbol{y} \in R^{300}$. The overall functions of sub-DNNs are denoted as $\varphi_x(\boldsymbol{x}) = g_3(\boldsymbol{\Psi}_3 \cdot g_2(\boldsymbol{\Psi}_2 \cdot g_1(\boldsymbol{\Psi}_1\boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2) + \boldsymbol{b}_3)$, where $\boldsymbol{\Psi}_i$ and $\boldsymbol{b}_i$ are the weight matrix and bias for the ith layer and $g_i(\cdot)$ is the activation function. And $\varphi_y(\boldsymbol{y})$ is computed in a similar way. Then, $\varphi_x(\boldsymbol{x})$ is the overall result of the CNN (or RNN) and its subsequent DNN, given the input spectrogram $\boldsymbol{s}$.

3.4 Objective Function of CCA

CCA [13] has been a very popular method for embedding multimodal data in a shared space, and is used to analyze the correlation between audio and lyrics here.

Assume the batch size in the training is N, $\boldsymbol{X} \in R^{D\times N}$ and $\boldsymbol{Y} \in R^{D\times N}$ are the outputs of sub DNN in one batch, corresponding to audio ($\varphi_x(\boldsymbol{x})$) and lyrics ($\varphi_y(\boldsymbol{y})$), respectively. Let covariance of $\varphi_x(\boldsymbol{x})$ and $\varphi_y(\boldsymbol{y})$ be $\boldsymbol{C}_{XX}$, $\boldsymbol{C}_{YY}$ and their cross-covariance be $\boldsymbol{C}_{XY}$. With the linear projection matrices $\boldsymbol{W}_X$ and $\boldsymbol{W}_Y$, the correlation between the canonical components ($\boldsymbol{W}_X^T\boldsymbol{X}$ and $\boldsymbol{W}_Y^T\boldsymbol{Y}$) can be computed by CCA. This correlation indicates the association between the two modalities and is

[3]https://developer.spotify.com/web-api/get-audio-features/
[4]https://ibm.ent.box.com/s/9ebs3c759qqo1d8i7ed323i6shv2js7e

used as an overall objective function, which is maximized to find all parameters (convolutional kernels or RNN parameters $\boldsymbol{H}(\cdot)$, non-linear projections $\varphi_x(\cdot)$ and $\varphi_y(\cdot)$, linear projection matrices $\boldsymbol{W}_X$ and $\boldsymbol{W}_Y$).

$$(\boldsymbol{H},\boldsymbol{W}_X,\boldsymbol{W}_Y,\varphi_x,\varphi_y) = \underset{(\boldsymbol{H},\boldsymbol{W}_X,\boldsymbol{W}_Y,\varphi_x,\varphi_y)}{argmax} corr(\boldsymbol{W}_X^T\boldsymbol{X},\boldsymbol{W}_Y^T\boldsymbol{Y}).$$

At first, with $\boldsymbol{H}, \varphi_x, \varphi_y$ being fixed, $\boldsymbol{W}_X$ and $\boldsymbol{W}_Y$ are computed by

$$(\boldsymbol{W}_X,\boldsymbol{W}_Y) = \underset{(\boldsymbol{W}_X,\boldsymbol{W}_Y)}{argmax} \frac{\boldsymbol{W}_X^T\boldsymbol{C}_{XY}\boldsymbol{W}_Y}{\sqrt{\boldsymbol{W}_X^T\boldsymbol{C}_{XX}\boldsymbol{W}_X \cdot \boldsymbol{W}_Y^T\boldsymbol{C}_{YY}\boldsymbol{W}_Y}}.$$

This can be rewritten in the trace form

$$\begin{aligned}(\boldsymbol{W}_X,\boldsymbol{W}_Y) &= \underset{(\boldsymbol{W}_X,\boldsymbol{W}_Y)}{argmax}\, tr(\boldsymbol{W}_X^T\boldsymbol{C}_{XY}\boldsymbol{W}_Y), \\ \text{subject to}: \quad & \boldsymbol{W}_X^T\boldsymbol{C}_{XX}\boldsymbol{W}_X = \boldsymbol{W}_Y^T\boldsymbol{C}_{YY}\boldsymbol{W}_Y = \boldsymbol{I}.\end{aligned} \tag{2}$$

Here, covariance $\boldsymbol{C}_{XX}$, $\boldsymbol{C}_{YY}$ and cross-covariance $\boldsymbol{C}_{XY}$ are computed as follows

$$\boldsymbol{C}_{XX} = \frac{1}{N-1}\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^T + r\boldsymbol{I}, \tag{3}$$

$$\boldsymbol{C}_{YY} = \frac{1}{N-1}\hat{\boldsymbol{Y}}\hat{\boldsymbol{Y}}^T + r\boldsymbol{I}, \tag{4}$$

$$\boldsymbol{C}_{XY} = \frac{1}{N-1}\hat{\boldsymbol{X}}\hat{\boldsymbol{Y}}^T, \tag{5}$$

$$\hat{\boldsymbol{X}} = \boldsymbol{\varphi_x(x)} - \overline{\boldsymbol{\varphi_x(x)}}, \hat{\boldsymbol{Y}} = \boldsymbol{\varphi_y(y)} - \overline{\boldsymbol{\varphi_y(y)}},$$

where $\overline{\boldsymbol{\varphi_x(x)}}$ and $\overline{\boldsymbol{\varphi_y(y)}}$ are the averages of $\varphi_x(\boldsymbol{x})$ and $\varphi_y(\boldsymbol{y})$ within the batch, and r is a small positive constant used to ensure the positive definiteness of $\boldsymbol{C}_{XX}$ and $\boldsymbol{C}_{YY}$.

By defining $\boldsymbol{T} \triangleq \boldsymbol{C}_{XX}^{-1/2}\boldsymbol{C}_{XY}\boldsymbol{C}_{YY}^{-1/2}$ and performing singular value decomposition (SVD) on $\boldsymbol{T}$ as $\boldsymbol{T} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, $\boldsymbol{W}_X$ and $\boldsymbol{W}_Y$ can be computed by [2]

$$\boldsymbol{W}_X = \boldsymbol{C}_{XX}^{-1/2}\boldsymbol{U}, \boldsymbol{W}_Y = \boldsymbol{C}_{YY}^{-1/2}\boldsymbol{V}. \tag{6}$$

Then, the item to be maximized in Eq.(2) can be rewritten as

$$tr((\boldsymbol{W}_X^T\boldsymbol{C}_{XY}\boldsymbol{W}_Y)^T \cdot \boldsymbol{W}_X^T\boldsymbol{C}_{XY}\boldsymbol{W}_Y) = tr(\boldsymbol{T}^T\boldsymbol{T}). \tag{7}$$

Accordingly, the gradient of the correlation with respect to $\hat{\boldsymbol{X}}$ is given by

$$\frac{1}{N-1}(2\nabla_{XX}\hat{\boldsymbol{X}} + \nabla_{XY}\hat{\boldsymbol{Y}}), \tag{8}$$

$$\nabla_{XX} = -\frac{1}{2}\boldsymbol{C}_{XX}^{-1/2}\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{C}_{XX}^{-1/2},$$

$$\nabla_{XY} = \boldsymbol{C}_{XX}^{-1/2}\boldsymbol{U}\boldsymbol{V}^T\boldsymbol{C}_{YY}^{-1/2}.$$

And the gradient of the correlation with respect to $\hat{\boldsymbol{Y}}$ can be computed in a similar way.

Then, the gradients are back propagated, first in the sub DNN, where $\varphi_x(\boldsymbol{x})$ and $\varphi_x(\boldsymbol{y})$ are updated. As for the audio branch, the gradients are further back propagated to the CNN/RNN layers, and the parameter $\boldsymbol{H}$ is updated.

4 EXPERIMENTS

The performances of the proposed DCCA variants are evaluated and compared with some baselines such as CCA and the deep multi-view embedding approach [9], using the cross-modal retrieval task.

This task is to retrieve lyrics with music audio as input or vice versa. With a given input (either audio slice or lyrics), its canonical component is computed, and its similarity with the canonical components of the other modality in the database is computed using the cosine similarity metric, and the results are ranked in the decreasing order of the similarity score.

4.1 Experiment Setting

Proposed methods. As discussed in Sec. 3, two variants of DCCA are investigated: 1) JT-CNN-DCCA (joint training of CNN and DCCA), 2) JT-RNN-DCCA (joint training of RNN and DCCA to better capture temporal properties).

Baseline methods include some shallow correlation learning methods (without fully connected layers between feature extraction and CCA) such as 3) Spotify-CCA (which applies CCA on the 65-dimensional audio features provided by Spotify), 4) PreT-CNN-CCA (which applies CCA on the features extracted by the pretrained CNN model), and deep correlation learning methods such as 5) Spotify-MVE (Spotify feature with the deep MVE method), 6) PreT-CNN-MVE (pretrained CNN model with the MVE method), and 7) Spotify-DCCA. In all these methods, the lyrics branch uses the features extracted by the pretrained Doc2Vec model.

The deep multi-view embedding (MVE) method is implemented in a way similar to [9], where arbitrary mappings of two different views are embedded in the joint space based on considering matched pairs with minimal distance and mismatched pairs with maximal distance. Both branches share the same parameters (activation function, number of neurons and so on) and both have 3 fully connected layers (with 512, 256, and 128 neurons respectively). Batch normalization is used before each layer and tanh activation function is applied after each layer.

Audio-lyrics dataset. Currently, there is no large audio/lyrics dataset publicly available for cross-modal music retrieval. Therefore, we build a new audio-lyrics dataset. Spotify is a music streaming on-demand service, which provides access to over 30 million songs, where songs can be searched by various parameters such as artist, playlist, and genre. Users can create, edit, and share playlists on Spotify. Initially, we take 20 most frequent mood categories (aggressive, angry, bittersweet, calm, depressing, dreamy, fun, gay, happy, heavy, intense, melancholy, playful, quiet, quirky, sad, sentimental, sleepy, soothing, sweet) [38] as playlist seeds to invoke Spotify API. For each mood category, we find the top 500 popular English songs according to the popularity provided by Spotify, and further crawl 30s audio samples from Spotify. Lyrics are collected from Musixmatch, and the length is adjusted to roughly match the audio length. Altogether there are 10,000 pairs of audio and lyrics.

Evaluation metric. In the retrieval evaluation, we mainly use mean reciprocal rank 1 (MRR1, which is defined as the mean of the reciprocal value of the rank of the only relevant item) as the metric. Because there is only one relevant audio or lyrics, MRR1 is able to show the rank of the result. We also evaluate recall@N to see how often the relevant item is included in the top N of the ranked list.

We use 8,000 pairs of audio and lyrics as the training dataset, and the rest 2,000 pairs for the retrieval testing. Because we generate 4 sub-sequences from each original MFCC sequence, there

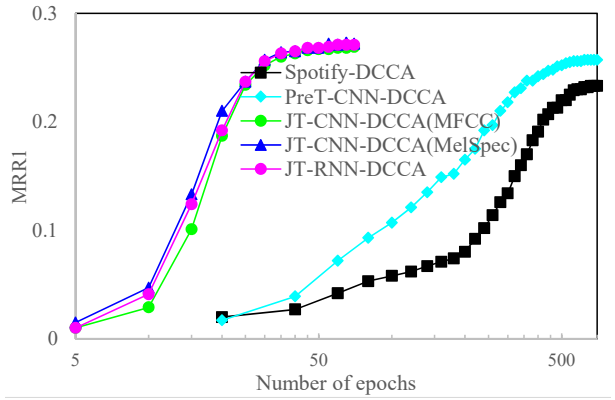

Fig. 3. MRR1 with respect to the numbers of epochs (Using audio as query to search lyrics, #CCA-component=30)

are 32,000 pairs of audio/lyric pairs for training in JT-CNN-DCCA with MFCC. In each run, the split of audio-lyrics pairs into training/testing is random, and a new model is trained. All results are averaged over 5 runs (cross-validations). In the batch-based training, the batch size is unified to 1000 samples in all methods, and the training takes 200 epochs for joint training and 700 epochs for other DCCA methods. Furthermore, training MVE requires the presence of non-paired instances. To this end, we randomly selected 1 non-paired instance for each song in the dataset. The margin hyper-parameter was set to 0.3, according to our preliminary experiments. Then, we trained MVE for 1280 epochs.

Experiment environment. The evaluations are performed on a Centos7.2 server, which is configured with two E5-2620v4 CPU (2.1GHz), three GTX 1080 GPU (11GB), and DDR4-2400 Memory (128G). Moreover, it contains CUDA8.0, Conda3-4.3 (python 3.5), Tensorflow 1.3.0, and Keras 2.0.5.

4.2 Performance under Different Numbers of Epochs

Fig. 3 shows the MRR1 results of Spotify-DCCA, PreT-CNN-DCCA, JT-CNN-DCCA with MFCC, JT-CNN-DCCA with Mel-spectrum, JT-RNN-DCCA with VGG16 features, under different numbers of epochs. In all methods, MRR1 increases with the number of epochs, but with different trends. It is clear that JT-CNN-DCCA(MFCC) and JT-RNN-DCCA have similar performance as JT-CNN-DCCA(MelSpec), converging much fast than the other two methods and achieving higher MRR1. Hereafter, we focus on JT-CNN-DCCA(MFCC) and JT-RNN-DCCA to investigate the performance of joint training.

4.3 Impact of Audio/Lyrics Length

The retrieval performance depends on the length of audio/lyrics. Here, we cut audio/lyrics into different lengths and evaluate how the retrieval performance changes. For the simplicity, here we only evaluate JT-RNN-DCCA and the MRR1 results, using either audio or lyrics as query, are shown in Fig. 4. The MRR1 results are obviously degraded at 5sec, but they are almost the same after 10sec. This reflects two facts as follows: (i) On one hand, lyrics and audios that can be correlated in the common canonical space match each other well even at relatively short length, and a long length is unnecessary. This is partly because of the repeat of lyrics/audio segments in the whole song. (ii) On the other hand, increasing the length has little effect on some lyrics/audios that are not so

Table 3. MRR1 with respect to different numbers of CCA/MVE components (Using audio as query)

#CCA	CCA		MVE		DCCA			
#MVE	Spotify	PreT-CNN	Spotify	PreT-CNN	Spotify	PreT-CNN	JT-CNN	JT-RNN
10	0.023	0.022	0.121	0.166	0.165	0.199	0.247	0.251
20	0.029	0.040	0.134	0.187	0.208	0.235	0.254	0.260
30	0.034	0.054	0.095	0.158	0.223	0.246	0.256	0.263
40	0.039	0.069	0.084	0.115	0.222	0.249	0.256	0.264
50	0.039	0.078	0.067	0.107	0.218	0.247	0.256	0.265
60	0.040	0.085	0.065	0.094	0.217	0.250	0.257	0.265
70	N/A	0.090	0.061	0.085	0.214	0.249	0.256	0.265
80	N/A	0.094	0.056	0.080	0.211	0.247	0.257	0.265
90	N/A	0.098	0.054	0.063	0.204	0.248	0.257	0.265
100	N/A	0.099	0.043	0.072	0.194	0.247	0.257	0.264

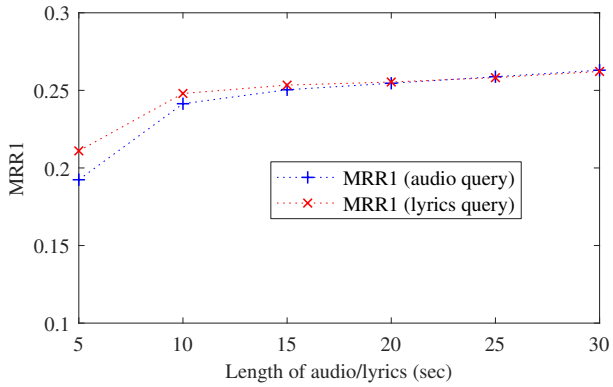

Fig. 4. MRR1 of JT-RNN-DCCA under different lengths of audio/lyrics (#CCA-component=30, 20% for testing)

similar. This is mainly due to the impact of background music, which affects the audio spectrum. This needs further investigation, and we leave it as future work.

4.4 Impact of Numbers of CCA Components

Here, we evaluate the impact of the number of CCA/MVE components, which affects the performance of both the baseline methods and the proposed methods. The number of CCA/MVE components is adjusted from 10 to 100. The results of MRR1 and recall of Spotify-CCA are marked as N/A when the number of CCA components is greater than 65, the dimension of Spotify feature.

The MRR1 results, with audio feature as query to search lyrics, are shown in Table 3. Clearly, with the linear CCA, Spotify-CCA and PreT-CNN-CCA have poor performance, although the performance increases with the number of CCA components. In comparison, with DCCA, the MRR1 results are much improved in Spotify-DCCA and PreT-CNN-DCCA. The MRR1 performance increases with the number of CCA components, and approaches a constant value in PreT-CNN-DCCA. MRR1 decreases a little in Spotify-DCCA when the number of CCA components gets greater than 65, the dimension of Spotify feature. Using MVE, the peak performance of Spotify-MVE and PreT-CNN-MVE lies between that of CCA and DCCA. With the end-to-end training, the MRR1 performance is further improved in JT-CNN-DCCA and reaches the maximum in JT-RNN-DCCA.

Table 4. Recall @ N with respect to different numbers of CCA components (Using audio as query)

	Spotify@1		PreT-CNN@1		JT-CNN@1	JT-RNN@1	Spotify@5		PreT-CNN@5		JT-CNN@5	JT-RNN@5
	CCA	DCCA	CCA	DCCA	DCCA	DCCA	CCA	DCCA	CCA	DCCA	DCCA	DCCA
10	0.006	0.134	0.007	0.170	0.233	0.234	0.025	0.190	0.025	0.227	0.257	0.262
20	0.010	0.178	0.020	0.214	0.243	0.247	0.034	0.213	0.047	0.253	0.262	0.269
30	0.014	0.195	0.031	0.227	0.245	0.251	0.043	0.245	0.068	0.262	0.263	0.271
40	0.019	0.195	0.045	0.231	0.245	0.252	0.047	0.245	0.085	0.265	0.262	0.272
50	0.020	0.190	0.053	0.230	0.246	0.253	0.049	0.240	0.095	0.260	0.262	0.272
60	0.020	0.191	0.060	0.232	0.246	0.253	0.051	0.237	0.102	0.264	0.263	0.273
70	N/A	0.187	0.065	0.232	0.246	0253	N/A	0.237	0.107	0.263	0.263	0.272
80	N/A	0.184	0.068	0.230	0.246	0.253	N/A	0.231	0.112	0.260	0.264	0.272
90	N/A	0.177	0.071	0.230	0.247	0.253	N/A	0.226	0.120	0.263	0.263	0.272
100	N/A	0.169	0.073	0.230	0.246	0.253	N/A	0.215	0.121	0.261	0.263	0.272

In these two methods, MRR1 is almost insensitive to the number of CCA components. But a further increase in the number of CCA components will lead to the SVD failure in CCA.

Table 4 shows the results of recall@1 and recall@5. Recall@5 in this table is only a little greater than recall@1, which indicates that for most queries, its relevant item either appears at the first place, or not in the top-N list at all. This infers that for some songs, lyrics and audio, even after being mapped to the same semantic space, are not similar enough.

4.5 Impact of Number of Training Samples

Here we investigate the impact of the number of training samples, by adjusting the percentage of samples for training from 20% to 80%. The percentage of samples for the retrieval test remains 20%, and the number of training samples is chosen in such a way that there are the same number of songs per mood category.

Fig. 5 shows the MRR1 results under the audio query. Spotify-CCA and PreT-CNN-CCA do not benefit from the increase of the training samples. Spotify-MVE and PreT-CNN-MVE benefits a little. In comparison, when DCCA is used, the increase of training samples enables the system to learn more diverse aspects of audio/lyric features, and the MRR1 performance almost linearly increases. In the future, we will try to crawl more data for training a better model to improve the retrieval performance.

Based on the above results, it is clear that JT-RNN-DCCA is superior over JT-CNN-DCCA, which also outperforms other methods. But even with explicit sequence modeling via RNN, the superiority of JT-RNN-DCCA over JT-CNN-DCCA is not large. This is partially because the kernels in CNN also capture the local temporal property. Although RNN further captures the temporal property in large time scale, its gain is not large because JT-RNN-DCCA does not benefit much from the increase in the lyrics/audio length.

5 CONCLUSION

Understanding the correlation between different music modalities is very useful for content-based cross-modal music retrieval and recommendation. Audio and lyrics are most interesting aspects for storytelling music theme and events. In this paper, a deep correlation learning between audio and lyrics is proposed to understand music audio and lyrics. This is the first research for deep cross-modal correlation learning between audio and lyrics. Some efforts are made to give a deep study: i) An end-to-end convolutional DCCA is proposed to learn correlation between audio and

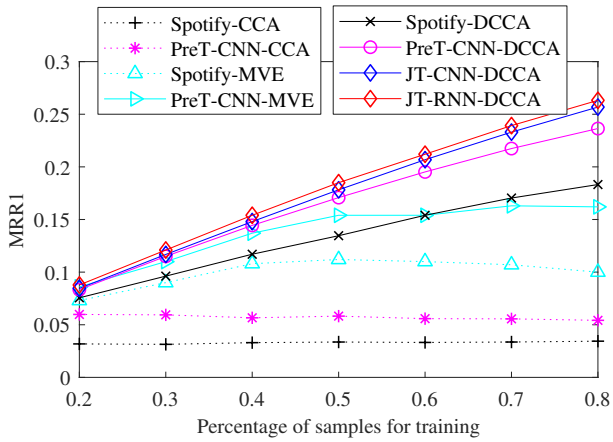

Fig. 5. MRR1 under different percentages of training samples (Using audio as query to search text lyrics, #CCA-component=30, 20% for testing)

lyrics where feature extraction and correlation learning are simultaneously performed and joint representation is learned by considering temporal structures. ii) An RNN network is studied for better learning temporal structures of music audio. iii) Extensive evaluations show the effectiveness of the proposed deep correlation learning architecture. More importantly, we apply our architecture to the bidirectional retrieval between audio and lyrics, e.g., searching lyrics with audio and vice versa.

This work mainly pays attention to studying deep models for processing music audio while using pre-trained Doc2Vec model for processing lyrics in the correlation learning. We are collecting more audio-lyrics pairs to further improve the retrieval performance, and will integrate music data in different modalities to implement personalized music recommendation. In the future work, we will investigate some deep models for processing lyrics branch. Lyrics contain a hierarchical composition such as verse, chorus, bridge. We will extend our deep architecture to complement musical composition (given music audio) where LSTM will be applied for learning lyrics dependencies.

REFERENCES

[1] Esra Acar, Frank Hopfgartner, and Sahin Albayrak. 2014. Understanding Affective Content of Music Videos Through Learned Representations. In *Proceedings of the 20th Anniversary International Conference on MultiMedia Modeling - Volume 8325 (MMM 2014)*. 303–314.

[2] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep Canonical Correlation Analysis. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28 (ICML'13)*. III–1247–III–1255.

[3] Eric Brochu, Nando de Freitas, and Kejie Bao. 2003. The Sound of an Album Cover: Probabilistic Multimedia and Information Retrieval. In *Artificial Intelligence and Statistics (AISTATS)*.

[4] Yue Cao, Mingsheng Long, Jianmin Wang, and Shichen Liu. 2017. Collective Deep Quantization for Efficient Cross-Modal Retrieval. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. 3974–3980.

[5] Keunwoo Choi, György Fazekas, and Mark B. Sandler. 2016. Automatic Tagging Using Deep Convolutional Neural Networks. In *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*. 805–811.

[6] Yandre M.G. Costa, Luiz S. Oliveira, and Carlos N. Silla. 2017. An Evaluation of Convolutional Neural Networks for Music Classification Using Spectrograms. *Appl. Soft Comput.* 52, C (2017), 28–38.

[7] Olivier Gillet, Slim Essid, and Gal Richard. 2007. On the Correlation of Automatic Audio and Visual Segmentations of Music Videos. *IEEE Transactions on Circuits and Systems for Video Technology* 17, 3 (2007), 347–355.

[8] Philippe Hamel, Matthew E. P. Davies, Kazuyoshi Yoshii, and Masataka Goto. 2013. Transfer Learning in MIR: Sharing Learned Latent Representations For Music Audio Classification And Similarity. In *ISMIR*.

[9] Wanjia He, Weiran Wang, and Karen Livescu. 2016. Multi-view Recurrent Neural Acoustic Word Embeddings. *CoRR* abs/1611.04496 (2016). http://arxiv.org/abs/1611.04496

[10] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. 2017. CNN architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 131–135.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[12] Sungeun Hong, Woobin Im, and Hyun Seung Yang. 2017. Deep Learning for Content-Based, Cross-Modal Retrieval of Videos and Music. *CoRR* abs/1704.06761 (2017). arXiv:1704.06761 http://arxiv.org/abs/1704.06761

[13] Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika* 28, 3/4 (1936), 321–377.

[14] Yan Huang, Wei Wang, and Liang Wang. 2017. Instance-aware Image and Sentence Matching with Selective Multimodal LSTM. In *IEEE CVPR'17*. 2310–2318.

[15] Qing-Yuan Jiang and Wu-Jun Li. 2017. Deep Cross-Modal Hashing. In *IEEE CVPR'17*. 3232–3240.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 60, 6 (2017), 84–90.

[17] Jey Han Lau and Timothy Baldwin. 2016. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. *CoRR* abs/1607.05368 (2016).

[18] Honglak Lee, Yan Largman, Peter Pham, and Andrew Y. Ng. 2009. Unsupervised Feature Learning for Audio Classification Using Convolutional Deep Belief Networks. In *Proceedings of the 22Nd International Conference on Neural Information Processing Systems (NIPS'09)*. Curran Associates Inc., USA, 1096–1104. http://dl.acm.org/citation.cfm?id=2984093.2984217

[19] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. 2017. Sample-level Deep Convolutional Neural Networks for Music Auto-tagging Using Raw Waveforms. *CoRR* abs/1703.01789 (2017). arXiv:1703.01789 http://arxiv.org/abs/1703.01789

[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*. 740–755.

[21] Ricardo Malheiro, Renato Panda, Paulo Gomes, and Rui Pedro Paiva. 2016. Emotionally-Relevant Features for Classification and Regression of Music Lyrics. *IEEE Transactions on Affective Computing* PP, 99 (2016), 1–1.

[22] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*. 55–60.

[23] Rudolf Mayer. 2011. Analysing the Similarity of Album Art with Self-Organising Maps. In *Advances in Self-Organizing Maps - 8th International Workshop, WSOM 2011, Espoo, Finland, June 13-15, 2011. Proceedings*. 357–366.

[24] Matt McVicar, Tim Freeman, and Tijl De Bie. 2011. Mining the correlation between lyrical and audio features and the emergence of mood. In *12th International Society for Music Information Retrieval Conference, Proceedings (ISMIR '11)*. 783–788.

[25] Rada Mihalcea and Carlo Strapparava. 2012. Lyrics, Music, and Emotions. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12)*. 590–599.

[26] Loris Nanni, Yandre M.G. Costa, Alessandra Lumini, Moo Young Kim, and Seung Ryul Baek. 2016. Combining Visual and Acoustic Features for Music Genre Classification. *Expert Syst. Appl.* 45, C (2016), 108–117.

[27] Bruno Nettl. 2000. An ethnomusicologist contemplates universals in musical sound and musical culture. *In N. Wallin, B. Merker, and S. Brown, editors, The origins of music, MIT Press, Cambridge, MA* (2000), 463–472.

[28] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert R.G. Lanckriet, Roger Levy, and Nuno Vasconcelos. 2010. A New Approach to Cross-modal Multimedia Retrieval. In *ACM MM'10*. 251–260.

[29] Rajiv Ratn Shah, Yi Yu, and Roger Zimmermann. 2014. ADVISOR: Personalized Video Soundtrack Recommendation by Late Fusion with Heuristic Rankings. In *Proceedings of the 22nd ACM International Conference on Multimedia (MM '14)*. 607–616.

[30] Siddharth Sigtia and Simon Dixon. 2014. Improved music feature learning with deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 6959–6963.

[31] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[32] Alexandros Tsaptsinos. 2017. Lyrics-Based Music Genre Classification Using a Hierarchical Attention Network. *CoRR* abs/1707.04678 (2017).
[33] Xixuan Wu, Yu Qiao, Xiaogang Wang, and Xiaoou Tang. 2016. Bridging Music and Image via Cross-Modal Ranking Analysis. *IEEE Transactions on Multimedia* 18, 7 (2016), 1305–1318.
[34] Fei Yan and Krystian Mikolajczyk. 2015. Deep correlation for matching images and text. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*. 3441–3450.
[35] Yi Yu, Michel Crucianu, Vincent Oria, and Lei Chen. 2009. Local Summarization and Multi-level LSH for Retrieving Multi-variant Audio Tracks. In *Proceedings of the 17th ACM International Conference on Multimedia (MM '09)*. 341–350.
[36] Yi Yu, Michel Crucianu, Vincent Oria, and Ernesto Damiani. 2010. Combining Multi-probe Histogram and Order-statistics Based LSH for Scalable Audio Content Retrieval. In *Proceedings of the 18th ACM International Conference on Multimedia (MM'10)*. 381–390.
[37] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. 2016. Video Captioning and Retrieval Models with Semantic Attention. *CoRR* abs/1610.02947 (2016).
[38] Yi Yu, Zhijie Shen, and Roger Zimmermann. 2012. Automatic Music Soundtrack Generation for Outdoor Videos from Contextual Sensor Information. In *Proceedings of the 20th ACM International Conference on Multimedia (MM '12)*. 1377–1378.
[39] Yi Yu, Suhua Tang, Kiyoharu Aizawa, and Akiko Aizawa. 2017. VenueNet: Fine-Grained Venue Discovery by Deep Correlation Learning. In *Proceedings of the 19th IEEE International Symposium on Multimedia (ISM'17)*. 3974–3980.
[40] Yi Yu, Suhua Tang, Kiyoharu Aizawa, and Akiko Aizawa. 2018. Category-Based Deep CCA for Fine-Grained Venue Discovery from Multimodal Data. *IEEE Transactions on Neural Networks and Learning Systems* (2018). https://doi.org/10.1109/TNNLS.2018.2856253
[41] Yi Yu, Roger Zimmermann, Ye Wang, and Vincent Oria. 2013. Scalable Content-Based Music Retrieval Using Chord Progression Histogram and Tree-Structure LSH. *IEEE Transactions on Multimedia* 15, 8 (2013), 1969–1981.
[42] Chunlin Zhong, Yi Yu, Suhua Tang, Shin'ichi Satoh, and Kai Xing. 2017. *Deep Multi-label Hashing for Large-Scale Visual Search Based on Semantic Graph.* Springer International Publishing, 169–184.

Received February 2007; revised March 2009; accepted June 2009