COVER SONG IDENTIFICATION USING SONG-TO-SONG CROSS-SIMILARITY MATRIX WITH CONVOLUTIONAL NEURAL NETWORK

Juheon Lee[1,4]*, Sungkyun Chang*[2,4]*, Sang Keun Choe*[3,4] *and Kyogu Lee*[2,4]

[1]College of Liberal Studies, [2]Music & Audio Research Group,
[3]Department of Electrical and Computer Engineering, [4]Center for Superintelligence,
Seoul National University, 08826, Korea

ABSTRACT

In this paper, we propose a cover song identification algorithm using a convolutional neural network (CNN). We first train the CNN model to classify any non-/cover relationship, by feeding a cross-similarity matrix that is generated from a pair of songs as an input. Our main idea is to use the CNN output–the cover-probabilities of one song to all other candidate songs–as a new representation vector for measuring the distance between songs. Based on this, the present algorithm searches cover songs by applying several ranking methods: 1. sorting without using the representation vectors; 2. the cosine distance between the representation vectors; and 3. the correlation between the vectors. In our experiment, the proposed algorithm significantly outperformed the algorithms used in recent studies, by achieving a mean average precision (MAP) of 93.18% in a dataset consisting of 3,300 cover-pairs and 496,200 non-cover-pairs.

Index Terms— Music Information Retrieval, Convolutional Neural Network, Cover Song Identification, Cross-similarity Matrix

1. INTRODUCTION

In music, cover songs include all kinds of songs which are reproduced by musicians other than the original producer or singer. These covers can share partial characteristics such as melody and lyrics with the original song, but they can also incorporate different musical elements such as the instrument(s) used, language of the lyrics, tempo, and key. The cover song identification study was conducted to determine whether the relationship between the two songs is covered or not, and to help prevent any copyright problem on the original song.

Previous studies on cover song identification are categorized by the information that is extracted from the song and how this information is compared [1]. The information for identifying the cover relationship is most often obtained by using tools such as chroma features [2, 3] and HPCPs (harmonic pitch class profiles) [4]. In addition, Euclidean distance, dynamic time warping(DTW), and SimPLe have been used for measuring the similarity between the features [5, 6].

In this study, we used a cross-similarity matrix between two chroma vectors as information for identification. From this, we designed a CNN [7, 8] to estimate the cover-probability of two songs. We used the probability values obtained from our CNN to represent the vectors of each song [9], and the cosine distance and correlation between them to represent the degree of similarity. The overview of the proposed system is shown in Fig. 1.

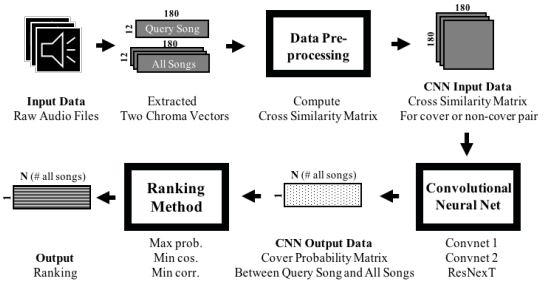


Fig. 1: System overview: a chroma vector is extracted from the audio of each song. Using the cross-similarity matrix derived from two chroma, the CNN outputs the cover-probability. Ranking is given based on the computed cover-probability matrix.

The paper is structured as follows. Section 2 explains how the cross-similarity matrix can identify certain characteristics when a cover relationship exists between two songs. Section 3 describes the structure of CNNs and provide detailed information on our proposed three models. Section 4 describes the ranking method that defines the distance between two songs with the probability values output from CNN, and computes ranking. We will then show the process and result of extracting the cover song identification performance index from the results of the learned CNN, and our overall findings, in Section 5. Finally, Section 6 will draw our conclusions.

2. CROSS-SIMILARITY MATRIX

To find the cover relationship between two songs, we have to compare the sub-melody sequence. Previous studies [5]

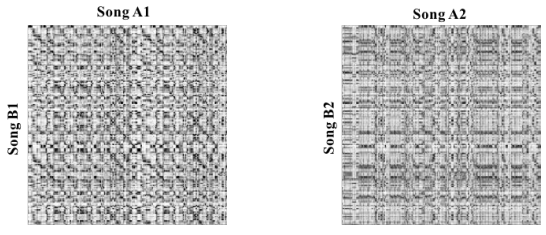
Table 1: Architecture of ConvNet-1: Inside the brackets are unit convolutional blocks, and outside the brackets is the number of stacked blocks. *Conv* denotes a *same* convolution layer with stride = 1, and its inside parentheses is (channel×width×height). *Maxpool* denotes a max-pooling layer with stride = 1, and its inside parentheses is (pooling size). ReLU, *BN*, *FC(the number of weights)*, and *DropOut(rate)* denote rectified linear unit activation function, batch normalization, fully-connected layer, and drop-out regularization, respectively.

Block #	Input layer	Block 1	Block 2	Block 3	Block 4	Block 5	Final layers	
Compo-nents	-	{ *Conv* (32 × 5 ×5), ReLU *Conv* (32 × 5 ×5), ReLU *Maxpool* (2 × 2) *BN* } × 1	{ *Conv* (32 × 3 ×3), ReLU *Conv* (16 × 3 ×3), ReLU *Maxpool* (2 × 2) *BN* } × 4				*DropOut*$_1$(0.5) *FC(256)*, ReLU *DropOut*$_2$(0.25)	*FC*(2) softmax
Output	(1, 180, 180)	(32,90,90)	(16,45,45)	(16,22,22)	(16,11,11)	(16,5,5)	(,,256)	(,,2)

Table 2: Architecture of ConvNet-2: Inside the brackets are unit convolutional blocks, and outside the brackets is the number of stacked blocks. *Conv* denotes a *valid* convolution layer with stride = 1, and we reuse the notations of Table 1.

Block #	Input layer	Block 1	Block 2	Block 3	Final layers	
Compo-nents	-	{ *Conv* (16 × 3 ×3), ReLU *BN* *Maxpool(2×2)* } × 2	{ *Conv* (32 × 3 ×3), ReLU *BN* *Maxpool(2×2)* } × 3	{ *Conv* (48 × 3 ×3), ReLU *Conv* (64 × 3 ×3), ReLU *Conv* (80 × 3 ×3), ReLU *Conv* (96 × 3 ×3), ReLU *Maxpool(2×2)* } × 1	*FC(1024)*, ReLU *DropOut*$_3$(0.5) *FC(200)*, ReLU *DropOut*$_4$(0.8)	*FC*(2) softmax
Output	(1,180,180)	(16,176,176)	(32,41,41)	(96,16,16)	(,,200)	(,,2)

mainly used chroma vectors [10] for this purpose. The 12-*d* chroma vectors extracted from the raw audio can represent the energy for 12 semi-tones per unit time. Using this chroma vectors, we compute a cross-similarity matrix to retrieve information about the relationship between the two songs. When extracting P and Q chroma vectors for song A and B each, the cross-similarity matrix has a size of $P \times Q$, and the Euclidean distance between the corresponding column and row chroma vector is input into each element. To avoid the key modulation between the two songs, we set the keys using an optimal transposition index (OTI) [11]. If the two songs are in cover-relationship, they will contain sections with similar melody lines. If the melody lines are similar, the chroma vectors obtained therefrom also will show similarities. Thus, in the cross-similarity matrix represented by the distance between the chroma vectors, a diagonal shape is consecutively formed, as shown in Fig. 2 (left). However, if two songs are not in a cover relationship, the probability that a similar melody lasts longer than a certain time is small, and then, to find a diagonal component is difficult, as shown in Fig. 2 (right). Based on this observations, we assume that the use of a CNN to detect Characteristic shape would be appropriate for identifying a cover relationship. Also, we observed that most of popular music recordings had durations of three to five minutes, and the first three minutes mostly contains main melodies. Thus, we assumed that the first 180 s of each song could provide relevant information to identify a cover song. Therefore, we used 180 seconds of each song to train network, and if the song lasted for less than 180 s, the duration of the song was standardized through zero-padding.



Song A1 : Always think of you (Kim Ran Young)
Song B1 : Always think of you (Lee Sun Hee)
Song A2 : Always think of you (Kim Ran Young)
Song B2 : The Magic Castle (Seo Young Eun)

Fig. 2: Example of cross-similarity matrices generated from cover song pair(left) and non-cover song pair(right): a diagonal component is found in the cross-similarity matrix extracted from the two songs in the cover relationship.

3. CNN FOR COVER SONG IDENTIFICATION

We build three types of CNNs for the cover song identification using a cross-similarity matrix: ConvNet-1 in Table 1, ConvNet-2 in Table 2, and ResNeXt [8]. These CNNs will be trained to output [1,0] when the two pieces of chroma information are similar, and [0,1] when they are not similar.

ConvNet-1 is a basic type of CNN with 10 convolutional layers containing 0.58×10^6 parameters. It initially subsamples the input with the filter size of 5×5; Convnet-2 is built as wider than ConvNet-1. It contains 25.28×10^6 parameters with 9 convolutional layers. It initially subsamples the input with the filter size of 3×3. Although these two CNNs have different block-architectures, every block in both nets basically outputs one-half down-sampled size of the input. The ResNeXt is an extended CNN that outperformed in the image classification task of ILSVRC 2016. Our implementation of ResNeXt followed the architecture and hyperparameters (cardinality = 32) presented in the paper [8].

4. RANKING METHOD

Assuming N songs, we can have $N \times N$ pairs of cross-similarity matrices as an input to the CNNs described in Section 3. Then, the output of CNNs produce a cover-probability matrix $P \in \mathbb{R}^{N \times N}$ where $P_{i,j}$ with $i, j \in \{1, 2, ..., N\}$ is

song index. Using P, we calculated the following ranking:

$$R_i^{\text{MaxProb}} = \text{sort}_{des}(P_{i,j} \text{ for all } j) \quad (1)$$

$R_i^{\text{MaxProb}} \in \mathbb{R}^{1\times N}$ is the ranking in which all the songs are arranged in order of probability of being in a cover relationship with the ith song. R_i^{MaxProb} can directly sort the values of $P_{i,j}$ for all $j \in \{1, 2, ..., N\}$ in descending order.

Now we propose another ranking methods that use the vectors derived from the P. If there were several songs similar to query song in the dataset, they can have similar representation vector components as the number of such songs. Therefore, they can get more robust close distance to query song when we use these kinds of ranking. We first represent the ith song with $P_{i,j}$ for all j, and then compute ranking based on the distance between the representation vectors. The representation vector for ith song $P_{i,:} \in \mathbb{R}^{1\times N}$ consists of the cover-probability values $P_{i,j}$ for all j. That is, $P_{i,:}$ has the form $[P_{i,1}, P_{i,2}, ..., P_{i,N}]$. Then, we define R_i^{MinCos} as :

$$R_i^{\text{MinCos}} = \text{sort}_{asc}(\text{dist}_{cos}(P_{i,:}, P_{j,:}) \text{ for all } j) \quad (2)$$

where dist_{cos} returns the cosine distance [12] between two vectors and sort_{asc} returns the index of elements sorted in ascending order. Then, in Eq. (2), R_i^{MinCos} computes the rankings in ascending order of cosine distance from the representation vector of the ith song. Additionally, We can replace the dist function in Eq. (2) with a correlation as :

$$R_i^{\text{MinCorr}} = \text{sort}_{asc}(\text{dist}_{corr}(P_{i,:}, P_{j,:}) \text{ for all } j) \quad (3)$$

where dist_{corr} is the correlation between two vectors [12].

Note that similar ranking methods can be found in previous studies [13, 14]. However, the present method differs from them in that we use the probability output (of CNN) as an element of the distance representation vector.

5. EVALUATION

5.1. Dataset

In Table 3, the train dataset consists of 1,175 songs: it has 2,113 cover pairs inside, and all other combinations are non-cover. The test dataset consists of 1,000 songs, and it resembles the dataset used in MIREX[1]: it has the same size with the dataset used in MIREX, and larger than that used in [15]. Note that these songs were originally collected by Heo, *et al* [14]. Of these, 30 kinds of 11 songs are covers of each other, totaling 330 songs. The rest of the 670 songs consists of songs that were not covered by any of the 1,000 songs except for the song itself. Train & test sets are disjoint from each other: there are no songs in both sets.

[1] http://www.music-ir.org/mirex/wiki/2016:Audio_Cover_Song_Identification

Table 3: Dataset information

Dataset	# cover	# non-cover
Train 2 K	2,113	2,113
Train 30 K	2,113	30,000
Train 100 K	2,113	100,000
Validation	322	322
Test	3,300	496,200

5.2. Metrics

We used three metrics of the same type, as used in the MIREX audio cover song identification task: **MNIT10** represents the average number of the songs that are true covers of the top ten songs judged to be covers for each song. **MAP**(mean average precision) is the average of the average precision values for each query song. **MR1**(mean rank 1) is mean rank of the first correctly identified as a cover.

5.3. Baseline algorithms

In the result of Section 5.4, DTW and SimPLe denote the *state-of-the-art* metric learning-based algorithm for audio cover song identification [14]. It takes the output of DTW or SimPLe as an input feature. To directly compare the performance of the present work with others, we used the baseline algorithm implemented by the authors [14].

5.4. Results

In experiment, we had three types of CNN model as {ConvNet-1, ConNet-2, ResNeXt}. Each of these CNNs was trained with the training sets of different sizes as {2 K, 30 K, 100 K}: we increased only the size of non-cover examples for investigating its effect on the cover song identification performance. Thus, we trained nine CNN models in total: We used Adam optimizer [16] with cross-entropy loss function [17], and the validation accuracy of each model was in the range of 0.83–0.88. We then applied three different types of ranking methods, proposed in section 4, to the output of each trained CNN. Finally, we evaluated the results of two baselines and 27 proposed algorithms, as displayed in Table 4.

In Table 4, the best MNIT10 of 9.16 (larger is better) and MAP of 0.93 were achieved by ConvNet-2+MinCorr using the largest training set size of 100 K. Also, ConvNet-1+MinCos and ConvNet-1+MinCorr trained with (training set size of) 30 K, ResNeXt+MinCos and ResNext+MinCorr trained with 100 K achieved the tie of best MAP. The best MR1 of 1.96 (smaller is better) was achieved by ResNeXt+ MaxProb trained with 30 K. In comparison with the recent metric learning-based algorithms (DTW+ML, SimPLe+ML), all the proposed models trained with 30 K or 100 K consistently outperformed as resulting up to 421 more number of

Model	Train set	ranking method	# correct answer	MNIT10	MAP	MR1
DTW+ML	-	MinEuclid	2406	7.29	0.75	26.55
SimPLe+ML	-		2602	7.88	0.81	15.05
ConvNet-1	2 K	MaxProb	2273	6.89	0.72	2.62
		MinCos	1657	5.02	0.52	16.5
		MinCorr	2090	6.33	0.67	8.67
	30 K	MaxProb	2655	8.05	0.83	2.50
		MinCos	3007	9.11	**0.93**	7.59
		MinCorr	3022	**9.16**	**0.93**	4.80
	100 K	MaxProb	2521	7.64	0.78	4.06
		MinCos	2888	8.75	0.89	8.32
		MinCorr	2911	8.82	0.90	10.7
ConvNet-2	2 K	MaxProb	1899	5.75	0.57	3.46
		MinCos	1621	4.91	0.52	10.5
		MinCorr	1852	5.61	0.60	8.01
	30 K	MaxProb	2647	7.99	0.82	2.92
		MinCos	2913	8.83	0.90	7.86
		MinCorr	2941	8.91	0.91	5.93
	100 K	MaxProb	2649	8.03	0.82	3.03
		MinCos	3008	9.12	0.92	10.3
		MinCorr	**3023**	**9.16**	**0.93**	7.01
ResNeXt	2 K	MaxProb	2525	7.65	0.77	2.90
		MinCos	1561	4.73	0.50	20.9
		MinCorr	1970	5.97	0.63	10.2
	30 K	MaxProb	2607	7.90	0.81	3.03
		MinCos	2955	8.95	0.91	3.26
		MinCorr	2954	8.95	0.91	2.87
	100 K	MaxProb	2705	8.20	0.84	**1.96**
		MinCos	3013	9.13	**0.93**	5.41
		MinCorr	3016	9.14	**0.93**	4.84

Table 4: Performance of cover song identification for baseline algorithms(DTW+ML, SimPLe+ML) and proposed algorithms. DTW + ML and SimPLe + ML are algorithms applying metric learning to DTW and SimPLe output values, respectively [14].

correct classifications, out of 3,300 ground-truth: our best result achieved 12.8 % point larger MNIT10, and 12% point larger MAP over than the SimPLe+ML algorithm. For MR1, the best performance was 15.05 in previous study, and it was remarkable that ResNeXt+MaxProb trained with 100 K, it could be lowered to 1.96. This revealed that the ground-truth cover songs appeared inside top 1.96 rank in average.

The effect of increasing non-cover examples for training: Overall, we could observe that the average MNIT10 and MAP of the models were improving in proportion to the size of non-cover examples for training, as shown in every case of ConvNet-2 and ResNeXt. A few exceptional cases were observed in the result of ConvNet-1, where the models trained with 30 K showed better average performance over than that trained with 100 K. In fact, ConvNet-1 had only 0.5×10^6 trainable parameters, while both ConvNet-2 and ResNext had more than 25×10^6 parameters. In this respect, the lack of network capacity could be thought as a plausible explanation for the case of ConvNet-1.

The effect of using vector representation for ranking: For the every model trained with 30 K or 100 K, we could observe that the average MNIT10 and MAP consistently improved when the proposed vector representation with MinCos or MinCorr was applied as a ranking method. In contrast, for the every model trained with 2 K, we could observe that the result of MaxProb was consistently better than vector representation-based ranking methods. One possible explanation for this issue could be made as follows. If the models were trained with 2 K, their output probabilities might not be as reliable as that of 30 K or 100 K. The comparably smaller MAP of the models with 2 K also reflects this. When using these lower-precision outputs as representation vectors, the noisy elements could be more dominant in calculation of rankings with cosine distance or correlation.

With respect to MR1, in contrast with MNIT10 and MAP, the proposed ranking method without using representation vector (MaxProb) outperformed over than other ranking methods. We could explain this issue as follows: if we have found only a small number of 10 covers for a query song, since the MaxProb method assigns a rank to a song that is determined to be cover even if it has small number, the MR1 is not greatly affected. However, if we use representation vector, the cosine distance or correlation value between the query song and the cover songs becomes relatively large, and therefore, MR1 performance could be lowered.

6. CONCLUSIONS AND FUTURE WORK

We proposed a novel approach for the cover song identification problem. Our prior observation was that the audio cover song relationship could appear as a diagonal component in the cross-similarity matrix. Based on this, we could train the CNNs to classify the cover and non-cover pair of songs by using the cross-similarity matrix as an image. Then, we proposed a ranking method using the output probability of the CNNs as a new representation vector for measuring the distance between songs. In experiment, we implemented three different types of CNNs: ConvNet-1, ConvNet-2, and ResNeXt. Within the dataset consisting of 3,300 cover-pairs and 496,200 non-cover-pairs of songs, the performance of the presented CNNs with the direct ranking method on the output probability was better than or comparable to that of *state-of-the-art*. After applying the proposed ranking method, we achieved 12 % point improvement of MAP over the direct ranking method. Additionally, we analyzed the effect of increased non-cover-examples in dataset for training, and the effect of using vector representation for the ranking method. Although this research showed promising results, we did not provide an entire framework for the large-scale search of cover songs. Future work will further investigate this issue.

7. ACKNOWLEDGEMENTS

This work was supported partly by Kakao and Kakao Brain corporations and partly by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2017M3C4A7078548).

8. REFERENCES

[1] Joan Serra, Emilia Gómez, and Perfecto Herrera, "Audio cover song identification and similarity: Background, approaches, evaluation, and beyond.," *Advances in Music Information Retrieval*, vol. 274, pp. 307–332, 2010.

[2] Kyogu Lee, "Identifying cover songs from audio using harmonic representation," *extended abstract submitted to Music Information Retrieval eXchange task, Victoria, BC, Canada*, 2006.

[3] Daniel PW Ellis and Graham E Poliner, "Identifying-cover songs' with chroma features and dynamic programming beat tracking," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, vol. 4, pp. IV–1429.

[4] Joan Serra and Emilia Gómez, "A cover song identification system based on sequences of tonal descriptors," *Music Information Retrieval Evaluation eXchange (MIREX)*, 2007.

[5] Joan Serra, Emilia Gómez, Perfecto Herrera, and Xavier Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1138–1151, 2008.

[6] Diego F Silva, Chin-Chin M Yeh, Gustavo Enrique de Almeida Prado Alves Batista, Eamonn Keogh, et al., "Simple: assessing music similarity using subsequences joins," in *International Society for Music Information Retrieval Conference, XVII*. International Society for Music Information Retrieval-ISMIR, 2016.

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

[8] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," *arXiv preprint arXiv:1611.05431*, 2016.

[9] Sungkyun Chang, Juheon Lee, Sang Keun Choe, and Kyogu Lee, "Audio cover song identification using convolutional neural network," *NIPS Machine Learning for Audio (ML4Audio) Workshop*, 2017.

[10] Ning Hu, Roger B Dannenberg, and George Tzanetakis, "Polyphonic audio matching and alignment for music retrieval," in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.* IEEE, 2003, pp. 185–188.

[11] Joan Serra, Emilia Gómez, and Perfecto Herrera, "Transposing chroma representations to a common key," in *IEEE CS Conference on The Use of Symbols to Represent Music and Multimedia Objects*, 2008, pp. 45–48.

[12] Sung-Hyuk Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, vol. 1, no. 2, pp. 1, 2007.

[13] Nathan N Liu and Qiang Yang, "Eigenrank: a ranking-oriented approach to collaborative filtering," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 83–90.

[14] Hoon Heo, Hyunwoo J Kim, Wan Soo Kim, and Kyogu Lee, "Cover song identification with metric learning using distance as a feature," *ISMIR*, 2017.

[15] Daniel PW Ellis and C Cotton, "The 2007 labrosa cover song detection system," *MIREX extended abstract*, 2007.

[16] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.