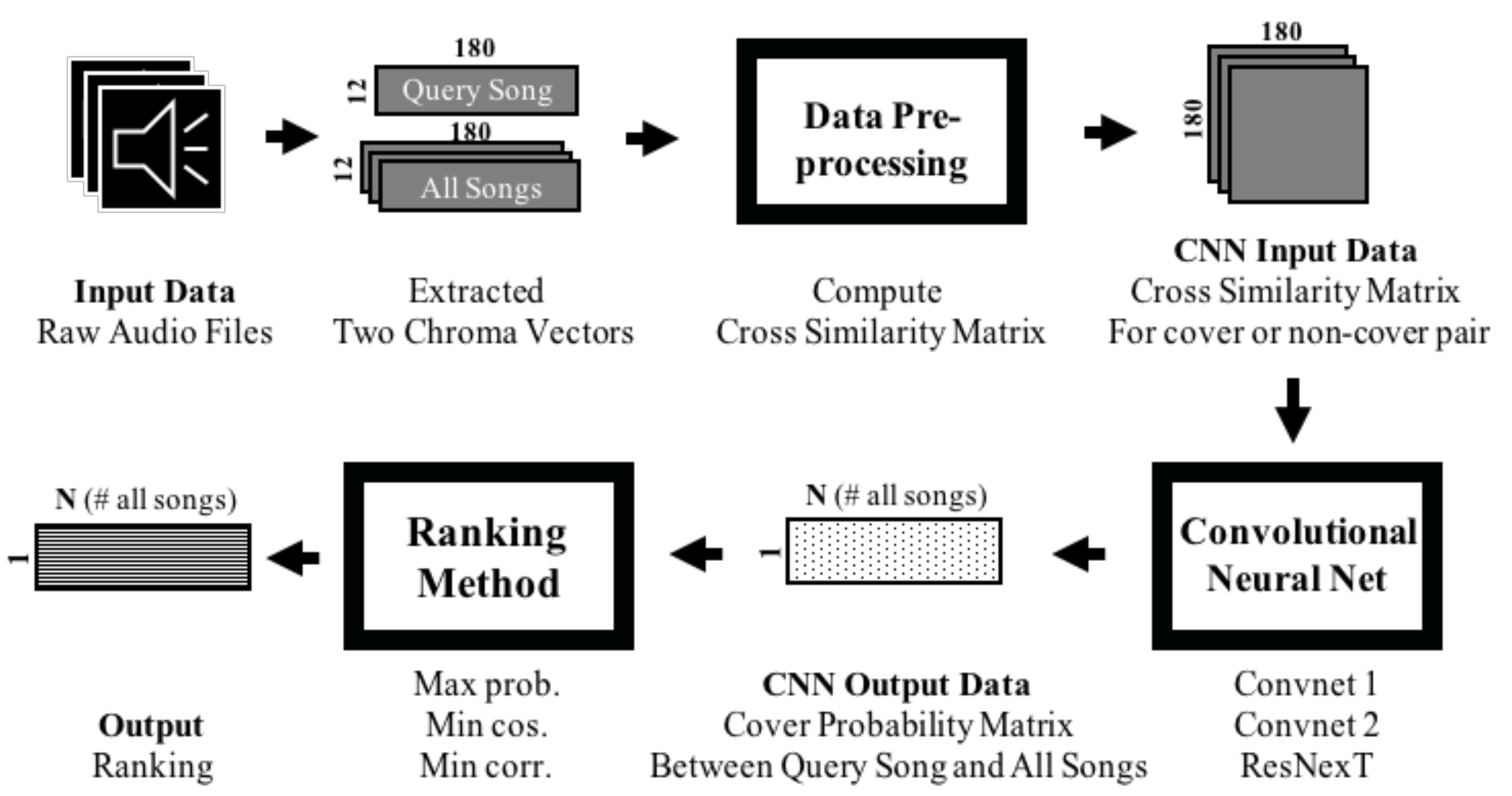
COVER SONG IDENTIFICATION USING SONG-TO-SONG CROSS-SIMILARITY MATRIX WITH CONVOLUTIONAL NEURAL NETWORK

Juheon Lee, Sungkyun Chang, Sangkeun Choe and Kyogu Lee
Music & Audio Research Group / Center for Super Intelligent
Seoul National University

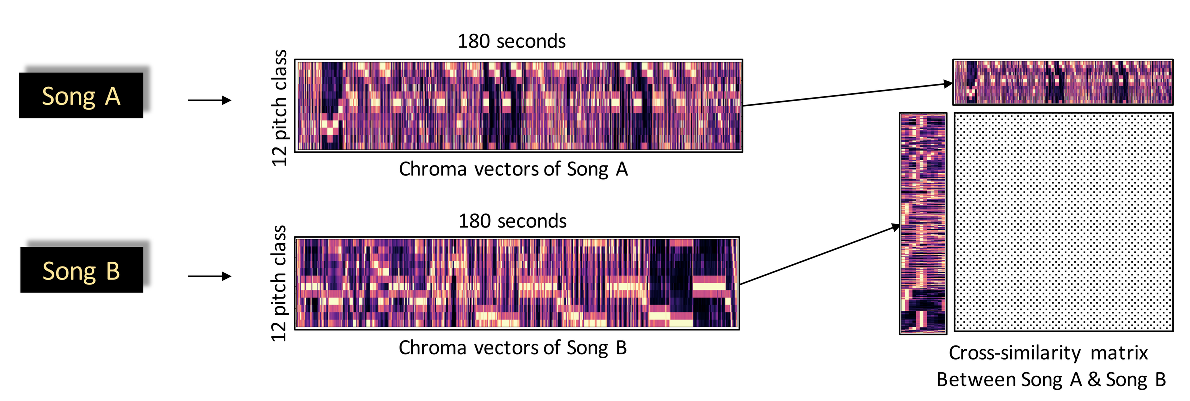
INTRODUCTION

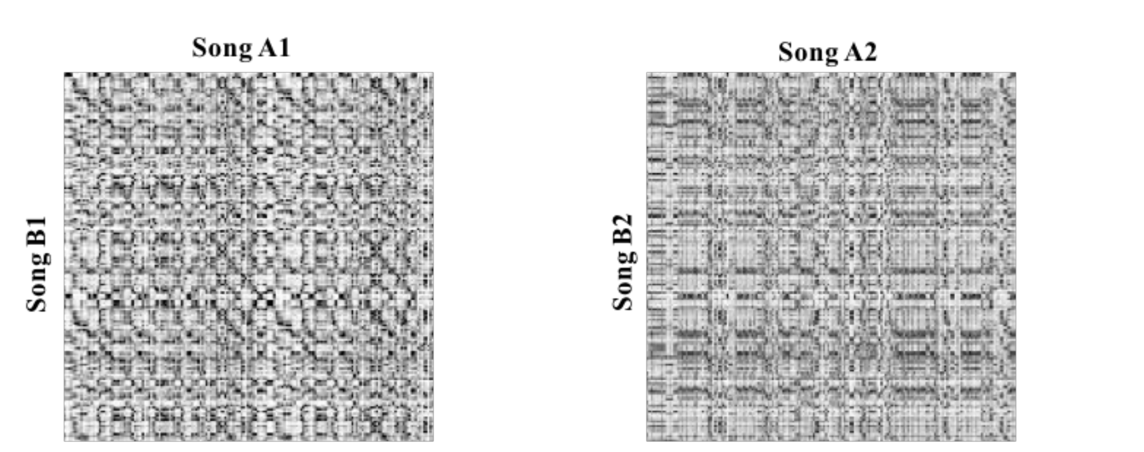
- **Cover Song identification** is a task that identifies songs that are covered by each other among various songs. This task contributes to the protection of intellectual property rights.
- The cover song shares a melody line similar to the original, but has differences in key, language, tempo, instruments, and so on.
- We propose a **CNN network with cross-similarity matrix** as a method to measure sub-sequence melody line similarity between cover and original song.
- We also propose a **cover song ranking method** based on the distance between the representation vectors composed of the cover-probabilities derived from CNN.

SYSTEM OVERVIEW



CROSS-SIMILARITY MATRIX





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)

- The *(m,n)* component of the cross-similarity matrix between song1 and song2 is the Euclidean distance between the *m-th* chroma vector of song1 and the *n-th* chroma vector of song2
- Apply OTI(Optimal Transpose Index) to avoid key modulation before compute cross-similarity matrix
- We assume that **black lines are formed diagonally** because they share similar sub-sequence melody lines when the two song are in cover-relationship.
- We designed the model in anticipation of CNN learning this diagonal orm.

ARCHITECTURE OF CNN

Block #	Input layer	Block 1	Block 2	Block 3	Block 4	Block 5	Final layers	
Compo-nents	-	$\left\{\begin{array}{c} Conv\ (32 \times 5 \times 5),\ \mathrm{ReLU} \\ Conv\ (32 \times 5 \times 5),\ \mathrm{ReLU} \\ Maxpool\ (2 \times 2) \\ BN \end{array}\right\} \times 1$	$\left\{\begin{array}{c} Conv\ (32 \times 3 \times 3),\ \mathrm{ReLU} \\ Conv\ (16 \times 3 \times 3),\ \mathrm{ReLU} \\ Maxpool\ (2 \times 2) \\ BN \end{array}\right\} \times 4$				$DropOut_1(0.5)$ $FC(256)$, 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)

[Convnet-1 structure]

Block #	Input layer	Block 1	Block 2	Block 3	Final layers	
Compo-nents	-	$\left\{\begin{array}{c} Conv\ (16 \times 3 \times 3),\ \mathrm{ReLU} \\ BN \\ Maxpool(2\times2) \end{array}\right\} \times 2$	$\left\{\begin{array}{c} Conv\ (32 \times 3 \times 3),\ \mathrm{ReLU} \\ BN \\ Maxpool(2\times2) \end{array}\right\} \times 3$	$\left\{\begin{array}{c} Conv\ (48 \times 3 \times 3),\ \mathrm{ReLU} \\ Conv\ (64 \times 3 \times 3),\ \mathrm{ReLU} \\ Conv\ (80 \times 3 \times 3),\ \mathrm{ReLU} \\ Conv\ (96 \times 3 \times 3),\ \mathrm{ReLU} \end{array}\right\} \times 1$ $Maxpool(2\times2)$	$FC(1024)$, 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)

[Convnet-2 structure]

- We constructed three simple CNN networks : **Convnet-1, Convnet-2, and ResNeXt :**

Convnet-1 is the common convolutional image identification network and it is composed of **narrow and deep structure**.

Convnet-2 is a **little wider than convnet1** and is designed to have more trainable parameters.

ResNeXt is a network that performs well in the image identification task, and we modified this network to suit the cover song identification task : such as input size, channel size, and so on.

RANKING METHOD

- To evaluate the cover song identification performance, we must calculate the rankings in the order of the highest probability that given query song and another song are in cover-relationship.
- Based on the cover-probability values output by CNN, we used three ranking methods to calculate the rank : MaxProb method / MinCos method / MinCorr method

1) MAX PROB Ranking Method

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

For the i-th song, calculate the cover-probability for all the other songs and assign a rank through descending order.

2) MIN COS Ranking Method

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

After assigning the cover-probability of a query song to all other songs as a representation vector, ranking is obtained from the order of shortest cosine distance between representation vectors.

3) MIN CORRELATION Ranking Method

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

After assigning the cover-probability of a query song to all other songs as a representation vector, ranking is obtained from the order of smallest correlation between representation vectors.

Cover Prob.	Song 1 (cover A)	Song 2 (cover A)	Song 3 (cover B)	Song 4 (cover B)	Song 5 (cover C)	Song 6 (cover C)	
Song 1 (cover A)	0.998 -	**0.978 Rank 1**	0.214 Rank 2	0.129 Rank 5	0.198 Rank 4	0.199 Rank 3	: Representation Vector of Song1 : [0.998, … , 0.199]
Song 2 (cover A)	**0.978 Rank 1**	0.997 -	0.110 Rank 3	0.087 Rank 4	0.032 Rank 5	0.126 Rank 2	: Representation Vector of Song2 : [0.978, … , 0.126]
Song 3 (cover B)	0.214 Rank 2	0.110 Rank 5	0.999 -	**0.966 Rank 1**	0.123 Rank 4	0.156 Rank 3	: Representation Vector of Song3 : [0.214, … , 0.156]
Song 4 (cover B)	0.129 Rank 2	0.087 Rank 4	**0.966 Rank 1**	0.967 -	0.089 Rank 3	0.067 Rank 5	: Representation Vector of Song4 : [0.129, … , 0.067]
Song 5 (cover C)	0.198 Rank 2	0.032 Rank 5	0.123 Rank 3	0.089 Rank 4	0.999 -	**0.879 Rank 1**	: Representation Vector of Song5 : [0.198, … , 0.879]
Song 6 (cover C)	0.199 Rank 2	0.126 Rank 4	0.156 Rank 3	0.067 Rank 5	**0.879 Rank 1**	0.987 -	: Representation Vector of Song6 : [0.199, … , 0.987]

Calculate cosine distance between rep.vectors

Cos dist.	Song 1	Song 2	Song 3	Song 4	Song 5	Song 6
Song 1	0.0 -	**0.011 Rank 1**	0.653 Rank 3	0.711 Rank 4	0.674 Rank 3	0.625 Rank 2
Song 2	**0.011 Rank 1**	0.0 -	0.733 Rank 2	0.787 Rank 4	0.796 Rank 5	0.740 Rank 3
Song 3	0.653 Rank 2	0.733 Rank 4	0.0 -	**0.004 Rank 1**	0.733 Rank 4	0.720 Rank3
Song 4	0.711 Rank 2	0.787 Rank 3	**0.004 Rank 1**	0.0 -	0.796 Rank 5	0.788 Rank 4
Song 5	0.674 Rank 2	0.796 Rank 4	0.733 Rank 3	0.796 Rank 4	0.0 -	**0.010 Rank 1**
Song 6	0.625 Rank 2	0.740 Rank 4	0.720 Rank 3	0,788 Rank 5	**0.010 Rank 1**	0.0 -

Calculate correlation between rep.vectors

Correlation	Song 1	Song 2	Song 3	Song 4	Song 5	Song 6
Song 1	0.0 -	**0.005 Rank 1**	1.499 Rank 5	1.497 Rank 4	1.456 Rank 3	1.416 Rank 2
Song 2	**0.005 Rank 1**	0.0 -	1.459 Rank 4	1.457 Rank 3	1.508 Rank 5	1.458 Rank 2
Song 3	1.499 Rank 3	1.459 Rank 2	0.0	**0.002 Rank 1**	1.512 Rank 4	1.552 Rank 5
Song 4	1.497 Rank 3	1.457 Rank 2	**0.002 Rank 1**	0.0	1.522 Rank 4	1.567 Rank 5
Song 5	1.456 Rank 2	1.508 Rank 3	1.512 Rank 4	1.522 Rank 5	0.0	**0.018 Rank 1**
Song 6	1.416 Rank 2	1.458 Rank 3	1.552 Rank 4	1.567 Rank 5	**0.018 Rank 1**	0.0

[Example of three ranking methods]

EXPERIMENT SETTING

DATASET we used 1175 pieces of Korean popular songs collected directly as a training set, and 1000 songs as a test set. There is no overlap between these two.

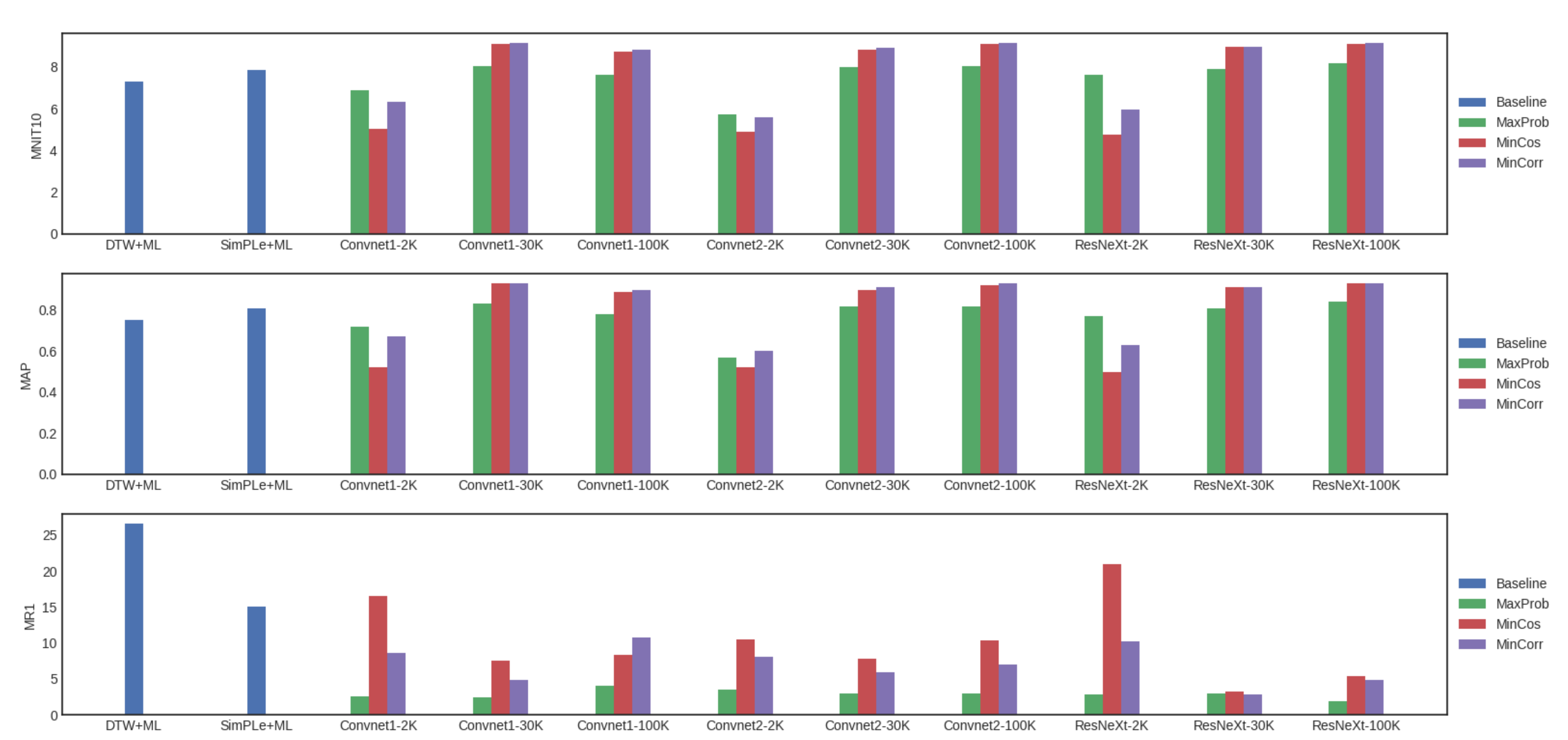
METRIC MNIT10 (mean number of covers identified in top 10), MAP (mean average precision), MR1(mean rank of the first correctly identified cover) are used as metrics.

BASELINE ALGORITHMS We set the state-of-the-art algorithm based on DTW, SimPLe, and metric learning algorithms as the baseline algorithm.

RESULT & DISCUSSION

Model	Train set	Ranking method	# correct answer	MNIT10	MAP	MR1
DTW+ML	-	MinEuclid	2046	7.29	0.75	26.55
SimPLe+ML	-	MinEuclid	2602	7.88	0.81	15.05
Convnet-1	30K	MinCorr	3022	9.16	0.93	4.80
Convnet-2	100K	MinCorr	3023	9.16	0.93	7.01
ResNeXt	100K	MaxProb	2705	8.20	0.84	1.96

[Best performance model in each metrics]



[all models performance compared to baseline algorithms]

: MNIT, MAP 15% better than the baseline algorithm.

The performance was higher when using 30K, 100K training sets than using 2K(# of non-cover pair). This is due to the fact that the network is learned properly in a real environment where there are more non-cover pairs than cover-pair. For the 30K, 100K training set, the performance was better when using the *MinCos* / *MinCorr* method than the *Maxprob* method. This is because the method uses the entire cover-probability value with all other songs as a representation vector, so that the probability of not making a correct judgment even if an error occurs in a specific cover-judgment becomes smaller.

: MR1 Improved to judge only two-times compared to judging the first cover song in fifteen-times.

Unlike the MNIT and MAP scores, the MR1 score showed the best performance when using the *MaxProb* method. This means that the ranking method that takes the highest value of the cover probability has a lower performance than the other methods when matching the top ten songs, but it is more advantageous to match a single definite cover song.

CONCLUSION

- We proposed cover song identification algorithm using CNN with cross-similarity matrix.
- The proposed algorithm with ranking method based on cover-probability perform 15% improvement over MNIT10 and MAP scores compared to the baseline algorithm.
- Our proposed algorithm also finds the cover song of the entire song for the first time in two attempts.

ACKNOWLEDGEMENT

This study was conducted with the support of KAKAO Brain.