COMBINING MUSICAL FEATURES FOR COVER DETECTION

Guillaume Doras[1,2] **Furkan Yesiler**[3]
Joan Serrà[4] **Emilia Gómez**[3,5] **Geoffroy Peeters**[6]
[1]Sacem, France [2]Ircam, CNRS, Sorbonne Université, STMS Lab, France
[3]Music Technology Group, Universitat Pompeu Fabra, Spain [4]Dolby Laboratories, Spain
[5]European Commission, Joint Research Centre, Spain [6]Telecom Paris, LTCI, France
guillaume.doras@ircam.fr, furkan.yesiler@upf.edu

ABSTRACT

Recent work have addressed the automatic cover detection problem from a metric learning perspective. They employ different input representations, aiming to exploit melodic or harmonic characteristics of songs and yield promising performances. In this work, we propose a comparative study of these different representations and show that systems combining melodic and harmonic features drastically outperform those relying on a single input representation. We illustrate how these features complement each other with both quantitative and qualitative analyses. We finally investigate various fusion schemes and propose methods yielding state-of-the-art performances on two publicly-available large datasets.

1. INTRODUCTION

Music retrieval has come a long way in the last 25 years. Since the earlier works on symbolic music retrieval [1, 2], applications with increasing complexity have been developed. In the mid-1990's, query-by-humming aimed at retrieving songs based on melodic similarity with a short hummed or whistled audio excerpt [3,4], while fingerprinting in the early 2000's aimed at identifying a song based on one of its excerpts [5]. Music matching at large – the task of retrieving an excerpt based on its musical similarity with another – was developed in the mid-2000's, typically comparing sequences of harmonic features via dynamic programming methods [6–8].

Automatic cover detection – the task of retrieving covers of a given track from an audio corpora – emerged at the same period, and was largely inspired by the previous decade of music retrieval research. Some of the early cover detection systems were relying on dominant melody to assess musical similarity [9, 10], and one of them reached the 3^{rd} place (out of 8 participants) at the first MIREX [1] cover song identification contest in 2006. The same year however, 1^{st} and 2^{nd} ranking algorithms were relying on harmonic features – chroma vectors or estimated chords series [11–13]. These results seem to have fostered the use of harmonic representations – chroma in particular – over melodic ones for cover detection, and all algorithms submitted to the next 2007 MIREX edition were using a tonal representation [14–16]. Enhanced chroma and time series comparison via dynamic programming then became the *de facto* standard method in the field – and remained the state of the art for more than a decade [17, 18].

During the following years, the community focused on improving both accuracy and scalability of existing approaches. As to accuracy, it was proposed to aggregate the results obtained with different methods [19–21] or different input features [22–24]. As to scalability, several strategies were investigated to compress the original representations and to reduce the similarity comparison function to a lightweight distance computation [25–28] or a fast lookup operation in a database index [29, 30].

The advent of efficient machine learning methods in other fields – such as image recognition – encouraged the community to shift from these previous methods based on ad-hoc and handcrafted features toward a new approach based on data-driven feature learning [31–33]. Recently, promising results were obtained using the metric learning paradigm in a cover detection context. The principle is to train a neural network to represent each track as a compact vector – its embedding – so that the distance between embeddings of a cover pair is smaller than that of non-cover pairs. Features used as input data were as varied as the Constant-Q Transform [34], dominant melody or multi-pitch [35, 36] or chord-informed chroma [37].

In this work, we propose a comparative study of these input features and investigate their combinations to improve cover detection performance. In Section 2, we briefly review different works inspiring our approach. In Section 3, we present the features that we consider for this study and their respective performances. In Section 4, we discuss the results obtained using different combinations of these features with a simple averaging fusion scheme, and explain them with a qualitative analysis. We then propose in Section 5 an architecture able to learn to combine various features efficiently. We conclude with future potential improvements.

[1] https://www.music-ir.org/mirex

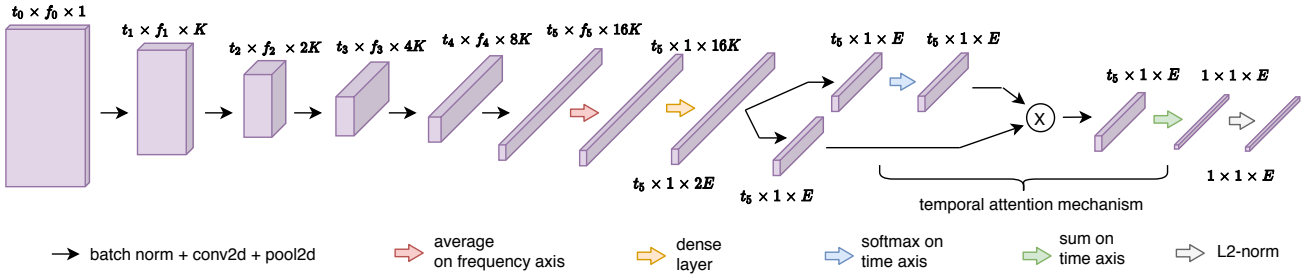


Figure 1: MICE architecture.

2. RELATED WORK

We present here the main concepts inspiring this work: input features combination and metric learning paradigm.

2.1 Combination of input features

A first attempt to combine information from various input features for cover detection was made by Foucard et al. using a source separation algorithm to obtain three inputs: the mixed original track, the dominant melody – assumed to correspond mainly to the solo singing voice – and the accompaniment [22]. In another study, Salamon et al. argued that, albeit closely related, dominant melody, bass line and harmonic progression embed different and complementary information. To prove this idea, they proposed to compare the systems that use each feature separately and their combinations with different fusion schemes [23]. More recently, Tralie et al. investigated another multi-representation approach, fusing harmonic and timbral features [24]. These studies showed that systems combining several input features outperformed those using each feature individually.

2.2 Classification vs. metric learning paradigm

Different teams recently proposed data-driven feature learning methods to address the cover detection problem. A common approach is to use a Convolutional Neural Network (CNN) to extract a compact representation – an embedding – from a low- or mid-level spectral representation of the audio, for instance Harmonic Pitch Class Profile (HPCP) [38] or Constant-Q Transform (CQT) [34, 39]. These authors considered the problem as a classification task, introducing an additional dense layer as a classifier.

Similarly, Doras et al. used dominant melody or multi-pitch representations [35, 40], while Yesiler et al. used crema-PCP [41], a chord-informed chroma representation [37] to extract the embedding. These input features were obtained with specialized neural networks [36, 41]. These authors also adopted a metric learning approach in which the CNN is trained with a triplet loss to produce embeddings whose pairwise Euclidean distance is lower for covers than for non-covers. Using these melodic or harmonic input features along with the metric learning paradigm yielded promising results and inspired this present work.

3. COMPARING INPUT FEATURES

We compare here performances obtained with a full spectral feature (CQT), two melodic features (dominant melody and multi-pitch) and two harmonic features (chroma and crema-PCP). For brevity, we denote them Cq, Dm, Mp, Ch, and Cp, respectively.

3.1 Input features

We computed Cq and Ch using Librosa v0.7 [42]. We obtained Dm and Mp with the convolutional network we previously described in [36], and we obtained Cp with the model publicly released by [41], as done in [37].

Temporal resolution All features were computed for an audio duration of 180 seconds as in [35], with a frame duration of 93ms (1937 bins). For tracks longer that 180 seconds, the beginning of the 180 seconds is taken at random, while shorter tracks were zero-padded, as in [37].

Frequency resolution All features were computed with a resolution of 1 bin per semi-tone. Cq was computed across 6 octaves. Dm and Mp are originally extracted with a resolution of 5 bins per semi-tone and their resolution is downsampled by a factor 5 via 2D-interpolation, following [35]. For each Dm, only the 3 octaves around its mean pitch are considered, as done in [35]. To account for all possible circular shifts in chroma features, we concatenate on top of the Ch and Cp their 11 lowest frequency bins, following [37, 38]. To summarize: Cq, Dm, Mp, Ch and Cp have 72, 36, 72, 23 and 23 frequency bins, respectively.

Normalization Cq and Ch are log-compressed and trimmed at -80dB. Finally, each feature is globally normalized between 0 and 1.

3.2 Model

Yesiler et al. introduced MOVE, a network containing a convolutional part specially designed to capture Cp patterns and a temporal attention part [37, 43], while Doras et al. used a plain convolutional network to capture Dm and Mp patterns [35, 40]. We introduce here a new model that reuses the plain convolutional part of the latter and the temporal attention mechanism of the earlier. The rationale behind this architecture is twofold: we need a generic model that can be used for all types of input features in order to conduct fair performance comparisons, and we observed in preliminary experiments that the temporal attention mechanism improves the results of the plain CNN. We call this model MICE (Musically Informed Cover Embeddings).

As shown on Figure 1, the first part of the model is the 5-layer CNN of [35]. Each layer block consists of a batch normalization layer, a convolution layer with 3×3 kernels and a mean-pooling layer with 2×2 kernel and 2×2 stride. The number of kernels K of the first layer is doubled at each level. Output is then averaged along the frequency axis, and a dense layer is applied to output a number of channels of $2E$, where E is the final embedding size.

The attention mechanism is then introduced: the tensor is split in 2 on its channels dimension to obtain two tensors of E channels. A softmax function is then applied on the time axis for the first tensor, and the output is multiplied element-wise with the second tensor. The resulting values are then summed along the time axis, which gives a vector of size E. The softmax followed by the multiplication and the sum implements a weighted average along the time axis per channel. The network is thus trained to give preference to the parts along the time dimension that are the most relevant to meet the objective function. The embedding vector is then L2-normalized. We used K=64 and E=512.

3.3 Experiments

In these first experiments, we train a different instantiation of MICE for each type of input feature and evaluate their cover detection performances.

Datasets We used the publicly available training set SHS_{5+} [2], containing Cq, Dm, Mp, Ch and Cp features for ~62k covers of ~7.5k works. It was split into a training/validation set with a ratio of 80/20 with respect to the works, i.e. all covers of a given work belong to one or the other set. We tested our model for each feature with two publicly available test datasets: SHS_{4-} [2], containing ~50k covers of ~20k works, and Da-TACOS [3], containing 13k covers of 1k works and 2k confusing tracks [44].

Loss We used a triplet loss to train this network [45]. Formally, if we let $\{a, p, n\}$ denote a triplet of track embeddings, where a is an anchor, and p or n is one of its covers or non-covers, respectively, the loss to minimize is expressed as $L = \max(0, d_{\text{ap}} + \alpha - d_{\text{an}})$, where α is a margin and d_{ap} and d_{an} are the distances between anchor a and p or n, respectively. We set $\alpha = 1$.

In practice, we used online semi-hard negative pairs mining [46], where triplets are built within each training batch: instead of using all possible triplets, each track in the batch is successively considered as an anchor, and compared with all its covers in the batch. For each of these positives pairs, if there are negatives such as $d_{an} < d_{ap}$, only the one with the highest d_{an} is kept. If no such negative exists, only the one with the lowest d_{an} is kept. Other negatives are not considered.

Training We train MICE with Adam optimizer [47], with an initial learning rate of $1e^{-4}$, divided by 2 each time the loss on the validation set has not decreased after 5k training steps. Training is stopped after 50k steps, or if the learning rate falls below $1e^{-7}$. The batch size is 64.

[2] https://gdoras.github.io/topics/coversdataset

[3] https://github.com/MTG/da-tacos

Testing For each feature, we use the corresponding trained model to compute the embeddings on the two test datasets. For SHS_{4-}, one cover per work is used as a query against the entire test set to compute a 20k×50k distance matrix. For Da-TACOS, each cover is used as a query against the entire dataset to compute a 13k×15k distance matrix. The Mean Average Precision (MAP), the mean number of correct answers in the ten first answers (MT@10) and the mean rank of first correct answer (MR1) are then computed.

3.4 Quantitative analysis

We report in Table 1 the performance scores obtained on Da-TACOS and SHS_{4-} for each type of input feature.

	Da-TACOS			SHS_{4-}		
Input	MAP	MT@10	MR1	MAP	MT@10	MR1
Cq	0.215	2.468	94	0.397	0.718	886
Dm	0.311	3.521	111	0.412	0.722	1431
Mp	0.293	3.290	**71**	0.422	0.760	**862**
Ch	0.121	1.476	117	0.174	0.371	1465
Cp	**0.375**	**4.084**	86	**0.499**	**0.842**	1169

Table 1: Results on SHS_{4-} and Da-TACOS for each feature.

Consistently, the Cp yields by far the best results, followed by the Dm and the Mp. This confirms our initial intuition that both melodic line and harmonic progression are prominent common musical facets between covers. Cq, representing the full spectrum, yields lower performance, which suggests that, albeit also embedded in the spectrum, the melodic and harmonic information is obfuscated, e.g. by percussive sounds information. Finally, the tonal information embedded in the Ch does not seem to be efficiently caught by our model.

From a practical point of view, crema-PCP is probably the best feature among those considered in this work, as it yields the best results with the lowest memory footprint.

4. COMBINING INPUT FEATURES

In this set of experiments, we now investigate if combining different features can improve the performance of each feature considered individually.

4.1 Are features complementary ?

We first compare pairwise embedding distances computed for the same pairs of tracks but obtained with different input features, as shown on Figure 2. The leftmost plot for instance compares the pairwise distances obtained for Dm and Cp. If each track's embeddings extracted from different input features were carrying the same information, the pairwise distance would be the same for a given pair of tracks, independently of the feature used. Figure 2 shows on the contrary that the same pair of tracks can obtain a low distance when using a given input feature, but a notably higher distance when using another one.

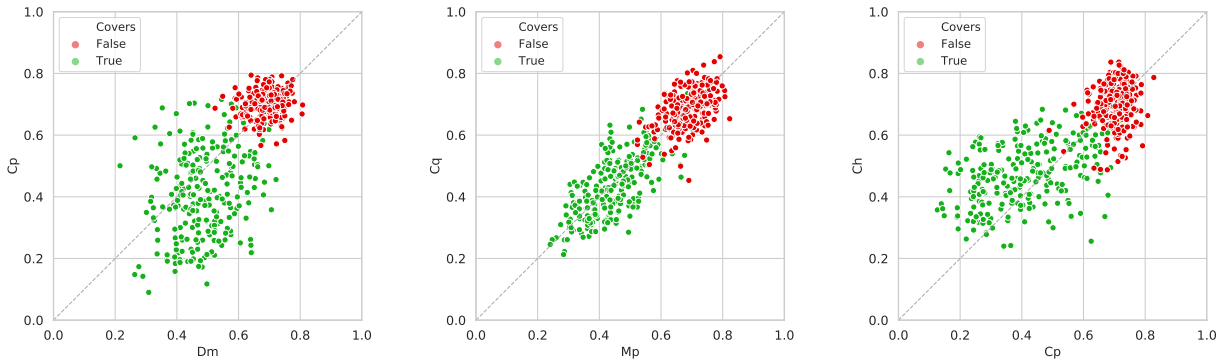


Figure 2: Comparison of the normalized distance obtained for the same pairs from SHS_{4-} (cover pairs in green and non-cover pairs in red) with different features: Dm vs. Cp (left), Mp vs. Cq (middle), Cp vs. Ch (right). Other combinations are not shown due to space constraints. For clarity, only 500 pairs randomly picked are drawn (250 covers and 250 non-covers).

All features seem relatively consistent when labeling non-cover pairs (red points exhibit high distances on both axes). Conversely, cover pairs (green points) are more scattered. Dm and Cp in particular seem to give very distinct results, as many pairs are spread far from the diagonal, which means that some cover pairs are more efficiently scored by one or the other feature. Intuitively, it seems logical that Dm and Cp are encoding complementary melodic and harmonic facets. This suggests that combining these features could benefit of this complementarity. We now conduct a quantitative and a qualitative analysis to confirm this intuition and to understand why certain representations yield better results for certain songs and vice-versa.

4.2 Quantitative analysis

We first experiment with a simple fusion scheme, which consists of averaging the pairwise distances obtained for the same pair with different features. We then re-compute the evaluation metrics based on this new averaged distance matrix for each possible feature combination. The rationale behind this approach is that we expect pairs incorrectly clustered with one representation to benefit from the correct clustering obtained with another representation.

The results are summarized in Table 2 for all combinations of Cq, Dm, Mp and Cp representations (we did not consider Ch here). We also computed the scores obtained by an oracle, which always picks among the distances obtained for each feature the lowest (resp. highest) distance for positive (resp. negative) pairs.

It appears clearly that any combination yields a better performance than any feature isolated (see Table 1). It also appears that the combinations where the Cp is used yield higher scores than the others, which was expected as Cp alone was already obtaining the highest scores. But more interestingly, we observe that the best improvements are obtained when combining melodic and harmonic features, i.e. Dm+Cp or Mp+Cp. The Mp probably embeds some of the information also carried by the Cp, as the improvement is lower when combining Mp+Cp than Dm+Cp.

All in all, the combination Dm+Cp yields the best performances, and an improvement of 15%-20% compared to Dm or Cp alone. Considering a third feature along Dm+Cp (Cq or Mp) improves the results slightly further.

Test set	Da-TACOS			SHS_{4-}		
Input	MAP	MT@10	MR1	MAP	MT@10	MR1
Cq+Dm	0.359	4.002	62	0.590	0.982	567
Cq+Mp	0.324	3.603	62	0.530	0.909	623
Cq+Cp	0.427	4.636	46	0.621	1.024	581
Dm+Mp	0.394	4.347	61	0.571	0.956	614
Dm+Cp	**0.547**	**5.861**	**37**	**0.679**	**1.098**	**529**
Mp+Cp	0.496	5.330	40	0.627	1.034	593
Cq+Dm+Mp	0.403	4.434	51	0.624	1.030	498
Cq+Dm+Cp	0.524	5.640	36	**0.713**	**1.143**	**430**
Cq+Mp+Cp	0.480	5.184	40	0.660	1.078	505
Dm+Mp+Cp	**0.553**	**5.939**	**35**	0.702	1.133	453
Dm+Cp (O)	0.800	8.360	4	0.873	1.344	115
Cq+Dm+Cp (O)	0.881	9.072	1	0.935	1.419	51
Dm+Mp+Cp (O)	0.874	9.022	2	0.924	1.405	63

Table 2: Comparison on Da-TACOS and SHS_{4-} of input feature combinations. Results obtained with the embeddings produced by MICE architecture trained for each feature (O=Oracle).

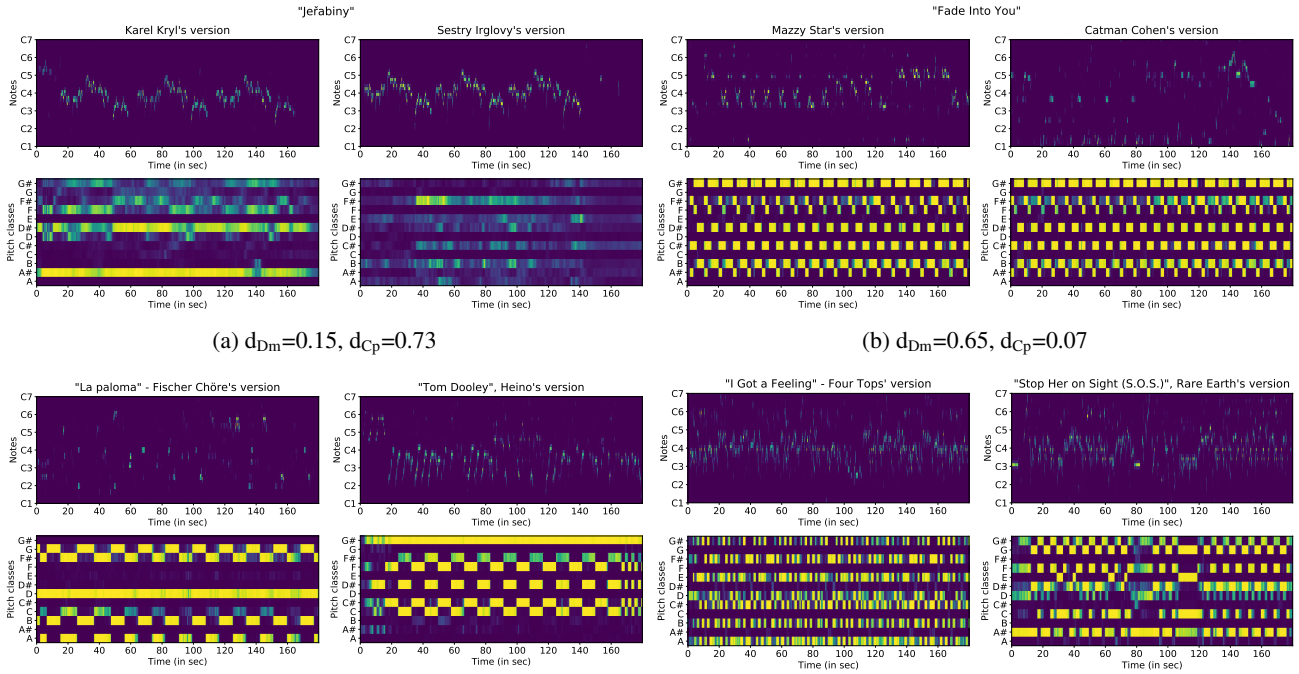
We also observe that the oracle scores about 20% above the highest scores obtained with the averaging fusion scheme, which suggests that further improvements are theoretically possible (we also experimented a minimum fusion scheme, which yielded lower performances).

From a practical perspective (e.g. memory footprint), the best trade-off seems to concentrate only on the Dm and the Cp. We will now investigate why the combination of these two features yields a better performance than others.

4.3 Qualitative analysis

To this aim, we selected the tracks where the first feature (e.g. Dm) gives particularly correct results and where the second feature (e.g. Cp) gives particularly incorrect results, or vice-versa. In other terms, we analyzed the pairs of songs for which the two features would give the most contradictory results for positive and negative pairs. The Dm and the Cp obtained for some of these cover and non-cover pairs [4] are shown on Figure 3.

[4] The audio of the songs described here can be listened on Youtube with the following IDs: Figure 3(a) clBw3cWgPnE and PNQeBX-tUdgc, Figure 3(b) -uJ61jgFCMM and xXvPFsoNnD4, Figure 3(c) 7nPBAiE76qY and bRrVMte9IQQ, Figure 3(d) pFrTXGEmU2Q and 3IOD9SqSfY4. Last accessed 11/5/2020.



(a) d_{Dm}=0.15, d_{Cp}=0.73

(b) d_{Dm}=0.65, d_{Cp}=0.07

(c) d_{Dm}=0.75, d_{Cp}=0.12

(d) d_{Dm}=0.22, d_{Cp}=0.84

Figure 3: Examples of cover pairs (top, (a) and (b)) and non-covers pairs (below, (c) and (d) where Dm and Cp gives contradictory results due to the melodic or harmonic content of each version. For each pair, Dm is displayed above and Cp below, and the corresponding distances d_{Dm} and d_{Cp} obtained for each feature are indicated.

Cover pairs Figure 3(a) displays two versions of "Jeřabiny", by Czech composer Karel Kryl (left, singing voice and guitar accompaniment) and Sestry Irglovy (right, purely a cappella, and poorly caught by the Cp). The pair is identified as covers thanks to the dominant melody.

Figure 3(b) displays two versions of "Fade Into You", by Mazzy Star (left) and Catman Cohen (right). The accompaniment is similar, but Catman Cohen's voice is very hoarse and rough, thus poorly caught by the dominant melody. The pair is identified as covers thanks to the Cp.

Non-cover pairs Figure 3(c) displays two different tracks: "La paloma" interpreted by a choir (left, mainly choir voices) and "Tom Dooley" by German singer Heino (right, voice and guitar accompaniment). Both songs share the same succession of two chords (but transposed), so the Cp are very similar. The pair is identified as non-covers thanks to the Dm, which are different.

Figure 3(d) displays two different tracks: "I Got a Feeling", by Four Tops (left) and "Stop Her on Sight (S.O.S.)" by Rare Earth (right). Both songs exhibit leading voice, backing voices, piano or strings section, and a brass instruments section. Both Dm appear very confused and look similar. The pair is identified as non-covers thanks to the Cp, which are different.

We could intuitively expect these results: Dm is better suited for songs where a melody is clearly prominent, while Cp is better suited for songs where no clear melody is present or is hidden by a prominent accompaniment. As such, there is no "better" feature: they simply perform differently on different tracks, and their combination performs statistically better on large corpora than separately.

5. LEARNING TO COMBINE FEATURES

Despite its encouraging results, the simple averaging fusion scheme has two flaws. Firstly, it does not guarantee that averaging the distances is the most optimal manner to merge the information contained in different representations. Many tracks might end up scoring around the mean of all distances, which will not help to decide whether they are covers or not. Secondly, it requires to train one model per representation, and consequently to store one embedding per representation, which complicates the operational usage of the system (e.g. indexing various embeddings and combining several search results is sub-optimal). In this section, we study the possibility to train a single model to learn how to fuse several input features efficiently.

We consider here only the combination of Dm and Cp features, as they individually yielded the most promising results with the averaging fusion scheme, while remaining practical from a memory footprint perspective.

5.1 Late fusion scheme

To address these flaws, we propose a two-branch architecture, where each input feature is processed by a dedicated model into an embedding, as previously. These two embeddings are then concatenated and merged into a single one by a final dense layer. We can now use different models for each feature, as we are not comparing their individual performance as previously. In particular, we use MOVE to process the Cp, as it was specially designed for this feature, and keep MICE to process the Dm, as shown on Figure 4.

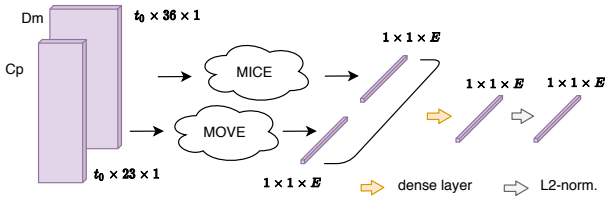


Figure 4: The late fusion architecture.

Let $\mathbf{e}_{\text{Dm}}$ and $\mathbf{e}_{\text{Cp}}$ denote the embeddings output by the Dm and Cp branches, $\mathbf{e}$ the final embedding and $\mathbf{W}$ the dense layer parameters. It comes:

$$\mathbf{e} = \mathbf{W} \begin{bmatrix} \mathbf{e}_{\text{Dm}} \\ \mathbf{e}_{\text{Cp}} \end{bmatrix} = \mathbf{W}^{\text{Dm}}\mathbf{e}_{\text{Dm}} + \mathbf{W}^{\text{Cp}}\mathbf{e}_{\text{Cp}} \quad (1)$$

where $\mathbf{W}^{\text{Dm}}$ and $\mathbf{W}^{\text{Cp}}$ are the parameters of $\mathbf{W}$ that are applied to $\mathbf{e}_{\text{Dm}}$ and $\mathbf{e}_{\text{Cp}}$, respectively. Normalizing $\mathbf{e}$ to unit norm, we can interpret the embedding resulting from this fusion as a weighted mean of the initial embeddings moved to another location on the unit sphere to optimize the loss.

5.2 Experiments

We compare here three training options: **a)** each branch and the last layer are trained simultaneously with random initialization from scratch; **b)** each branch is first pre-trained individually with its corresponding input feature as previously; then their trained weights are reloaded in the late fusion architecture, and are fine tuned while training the last layer; **c)** is the same as **b)**, but the weights of each branch are frozen once reloaded in the fusion model, and only the weights of the final dense layer are learned.

For these three options, we train each architecture on the same proprietary training set that was used in [37]. This set contains 98k tracks and is much larger than the one used in features comparison experiments of Section 4.2. Models trained with this proprietary training set were evaluated with Da-TACOS in order to compare the results with the baseline established in [37]. We also conduct the same experiments for each architecture trained on SHS_{5+} and evaluated with SHS_{4-} as previously, in order to compare the results with the baseline established in [40].

The training procedure for all three options is the same as described in Section 3.3, except that the learning rate is initialized at 5e^{-6} for option **b)** and at 1e^{-1} for option **c)**.

5.3 Results

The results of the late fusion learning experiments are summarized on Table 3. We indicated the scores obtained for each feature (Dm and Cp) individually, as well as the corresponding distance averaging score for comparison.

For both sets, the two-branch model outperforms the ones where each feature is considered individually, which shows that it jointly learns from both input features. Late fusion with end-to-end training from scratch (option **a)**) scores below the other two options, which suggests that the model learns from each feature but does not make an optimal use of the available information.

Late fusion with fine tuning of the pre-trained branches (option **b)**) yields better results. However, it does not outperform the late fusion where only the dense layer is trained (option **c)**). A possible explanation could be that one feature tends to yield better results than the other (probably the Cp, as seen in Section 3), and allowing the update of branches might confuse the previously acquired knowledge of the weaker one. This assumption should however be investigated further in another work.

	Da-TACOS			SHS_{4-}		
Input	MAP	MT@10	MR1	MAP	MT@10	MR1
Dm (MICE)	0.360	4.032	94	0.412	0.722	1431
Cp (MOVE)	0.484	5.214	59	0.533	0.890	1188
Dm+Cp (A)	0.621	6.613	32	**0.697**	**1.120**	**517**
Dm+Cp (LF-a)	0.570	6.101	**29**	0.617	1.017	686
Dm+Cp (LF-b)	0.592	6.318	32	0.655	1.059	655
Dm+Cp (LF-c)	**0.635**	**6.744**	30	0.660	1.080	657
Doras et al. [40]	n/a	n/a	n/a	0.323	0.615	1476
Yesiler et al. [37]	0.507	-	40	n/a	n/a	n/a

Table 3: Comparison on Da-TACOS (resp. SHS_{4-}) of all fusion schemes trained on [37] proprietary training set (resp. SHS_{5+}). A=averaging, LF=Late fusion. Note that Cp and Dm+Cp (A) scores are higher here than in Table 2 because Cp is now processed by MOVE.

Training a dense layer on top of two pre-trained frozen branches (option **c)**) thus yields the best scores, similar to the ones obtained by the averaging scheme.

We finally compare these performances to the current state of the art for each set. We observe that all late fusion schemes notably outperform the results obtained in [40] and [37], for the same training and test sets.

6. CONCLUSION

We proposed in this work a comparative study of different input features that have been used in recent works addressing the cover detection problem with a metric learning approach. We observed that the best feature of the one we studied is the crema-PCP, a harmonic feature. We then showed that combining this feature with a dominant melody representation drastically improves the results compared to each feature considered alone. We showed that this can be explained by the fact that using both melodic and harmonic features helps to disambiguate pairs of tracks that don't have a clear melodic or harmonic structure. We finally proposed a late fusion scheme learning to combine input features, which yields to new state-of-the-art performances on two publicly available datasets.

This system could be improved in several ways. As suggested by the oracle results, further strategies could be developed to force the model to focus more adequately on the available features. Also, the need to maintain several dedicated branches in the late fusion scheme could be avoided with a single architecture merging the two branches earlier in the process. But perhaps more importantly, considering the variety of other features commonly shared by covers, such as lyrics, could be a fruitful strategy to build future cover detection systems.

7. ACKNOWLEDGMENTS

FY is supported by the MIP-Frontiers project, the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 765068, and EG by TROMPA, the Horizon 2020 project 770376-2.

8. REFERENCES

[1] J. S. Downie, *Evaluating a simple approach to music information retrieval: Conceiving melodic n-grams as text*. Faculty of Graduate Studies, University of Western Ontario London, Ont., 1999.

[2] S. Doraisamy and S. M. Rüger, "An approach towards a polyphonic music retrieval system." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2001.

[3] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by humming," in *Proceedings of ACM Multimedia*, 1995.

[4] T. Nishimura, H. Hashiguchi, J. Takita, J. X. Zhang, M. Goto, and R. Oka, "Music signal spotting retrieval by a humming query using start frame feature dependent continuous dynamic programming." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2001.

[5] A. Wang, "An industrial strength audio search algorithm." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2003.

[6] J. P. Bello and J. Pickens, "A robust mid-level representation for harmonic content in music signals." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2005.

[7] M. Müller, F. Kurth, and M. Clausen, "Audio matching via chroma-based statistical features." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2005.

[8] F. Kurth and M. Müller, "Efficient index-based audio matching," *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

[9] W.-H. Tsai, H.-M. Yu, H.-M. Wang *et al.*, "Query-by-example technique for retrieving cover versions of popular songs with similar melodies." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2005.

[10] C. Sailer, "Using string alignment in a query-by-humming system for real world applications," *The Journal of the Acoustical Society of America*, 2005.

[11] E. Gómez and P. Herrera, "The song remains the same: identifying versions of the same piece using tonal descriptors." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2006.

[12] D. P. Ellis and G. Poliner, "Identifying 'cover songs' with beat-synchronous chroma features," in *MIREX (Music Information Retrieval Evaluation eXchange)*, 2006.

[13] K. Lee, "Identifying cover songs from audio using harmonic representation," *MIREX (Music Information Retrieval Evaluation eXchange)*, 2006.

[14] J. Serrà and E. Gómez, "A cover song identification system based on sequences of tonal descriptors," *MIREX (Music Information Retrieval Evaluation eXchange)*, 2007.

[15] D. P. Ellis and C. V. Cotton, "The 2007 labrosa cover song detection system," 2007.

[16] J. P. Bello, "Audio-based cover song retrieval using approximate chord sequences: Testing shifts, gaps, swaps and beats." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2007.

[17] J. Serrà, E. Gómez, P. Herrera, and X. Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

[18] J. Serrà, X. Serra, and R. G. Andrzejak, "Cross recurrence quantification for cover song identification," *New Journal of Physics*, 2009.

[19] S. Ravuri and D. P. Ellis, "Cover song detection: from high scores to general classification," in *Proceedings of ICASSP (International Conference on Acoustics, Speech and Signal Processing)*. IEEE, 2010.

[20] A. Degani, M. Dalai, R. Leonardi, and P. Migliorati, "A heuristic for distance fusion in cover song identification," in *International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*. IEEE, 2013.

[21] J. Osmalskyj, J.-J. Embrechts, P. Foster, and S. Dixon, "Combining features for cover song identification," in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2015.

[22] R. Foucard, J.-L. Durrieu, M. Lagrange, and G. Richard, "Multimodal similarity between musical streams for cover version detection," in *Proceedings of ICASSP (International Conference on Acoustics, Speech and Signal Processing)*. IEEE, 2010.

[23] J. Salamon, J. Serrà, and E. Gómez, "Melody, bass line, and harmony representations for music version identification," in *Proceedings of the 21st International Conference on World Wide Web*. ACM, 2012.

[24] C. J. Tralie, "Early mfcc and hpcp fusion for robust cover song identification," *arXiv preprint arXiv:1707.04680*, 2017.

[25] T. Bertin-Mahieux and D. P. Ellis, "Large-scale cover song recognition using the 2d fourier transform magnitude." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2012.

[26] D. F. Silva, C.-C. M. Yeh, G. E. d. A. P. A. Batista, and E. Keogh, "Simple: assessing music similarity using subsequences joins," in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2016.

[27] D. F. Silva, F. V. Falcao, and N. Andrade, "Summarizing and comparing music data and its application on cover song identification," in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2018.

[28] P. Seetharaman and Z. Rafii, "Cover song identification with 2d fourier transform sequences," in *Proceedings of ICASSP (International Conference on Acoustics, Speech and Signal Processing)*. IEEE, 2017.

[29] M. Marolt, "A mid-level representation for melody-based retrieval in audio collections," *IEEE Transactions on Multimedia*, 2008.

[30] P. Grosche and M. Müller, "Toward characteristic audio shingles for efficient cross-version music retrieval," in *Proceedings of ICASSP (International Conference on Acoustics, Speech and Signal Processing)*. IEEE, 2012.

[31] E. J. Humphrey, O. Nieto, and J. P. Bello, "Data driven and discriminative projections for large-scale cover song identification." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2013.

[32] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," Ph.D. dissertation, 2016.

[33] T. Tsai, T. Prätzlich, and M. Müller, "Known artist live song id: A hashprint approach." in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2016.

[34] Z. Yu, X. Xu, X. Chen, and D. Yang, "Learning a representation for cover song identification using convolutional neural network," *Proceedings of ICASSP (International Conference on Acoustics, Speech and Signal Processing)*, 2020.

[35] G. Doras and G. Peeters, "Cover detection using dominant melody embeddings," in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2019.

[36] G. Doras, P. Esling, and G. Peeters, "On the use of u-net for dominant melody estimation in polyphonic music," in *International Workshop on Multilayer Music Representation and Processing (MMRP)*. IEEE, 2019.

[37] F. Yesiler, J. Serrà, and E. Gómez, "Accurate and scalable version identification using musically-motivated embeddings," *Proceedings of ICASSP (International Conference on Acoustics, Speech and Signal Processing)*, 2020.

[38] X. Xu, X. Chen, and D. Yang, "Key-invariant convolutional neural network toward efficient cover song identification," in *Proceedings of IEEE ICME (International Conference on Multimedia and Expo)*. IEEE, 2018.

[39] Z. Yu, X. Xu, X. Chen, and D. Yang, "Temporal pyramid pooling convolutional neural network for cover song identification," *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019.

[40] G. Doras and G. Peeters, "A prototypical triplet loss for cover detection," in *Proceedings of ICASSP (International Conference on Acoustics, Speech and Signal Processing)*, 2020.

[41] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition," in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2017.

[42] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015.

[43] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series." in *CCIA*, 2018.

[44] F. Yesiler, C. Tralie, A. A. Correya, D. F. Silva, P. Tovstogan, E. Gómez, and X. Serra, "Da-tacos: A dataset for cover song identification and understanding," in *Proceedings of ISMIR (International Society for Music Information Retrieval)*, 2019.

[45] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in neural information processing systems*, 2006.

[46] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of IEEE CVPR (Conference on Computer Vision and Pattern Recognition)*, 2015.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.