DISCOGS-VI: A MUSICAL VERSION IDENTIFICATION DATASET BASED ON PUBLIC EDITORIAL METADATA

R. Oguz Araz **Xavier Serra** **Dmitry Bogdanov**
Music Technology Group, Universitat Pompeu Fabra, Barcelona
{recepoguz.araz, xavier.serra, dmitry.bogdanov}@upf.edu

ABSTRACT

Current version identification (VI) datasets often lack sufficient size and musical diversity to train robust neural networks (NNs). Additionally, their non-representative clique size distributions prevent realistic system evaluations. To address these challenges, we explore the untapped potential of the rich editorial metadata in the Discogs music database and create a large dataset of musical versions containing about 1,900,000 versions across 348,000 cliques. Utilizing a high-precision search algorithm, we map this dataset to official music uploads on YouTube, resulting in a dataset of approximately 493,000 versions across 98,000 cliques. This dataset offers over nine times the number of cliques and over four times the number of versions than existing datasets. We demonstrate the utility of our dataset by training a baseline NN without extensive model complexities or data augmentations, which achieves competitive results on the SHS100K and Da-TACOS datasets. Our dataset, along with the tools used for its creation, the extracted audio features, and a trained model, are all publicly available online.

1. INTRODUCTION

Artists continue to cover, remix, and reinterpret musical works, creating a rich tapestry of musical versions that celebrate the originals. This proliferation presents a complex challenge: how to accurately identify different versions of a musical work within vast digital catalogs. Version identification (VI) addresses this problem using audio processing methods to find versions of query tracks in music catalogs [1–3]. VI has thus emerged as a crucial solution with significant implications across multiple applications including music discovery, musicological research, and copyright enforcement. From both the artists' and copyright holders' perspectives, VI has substantial importance as it offers a tool for financial compensation to many music industry stakeholders.

Recently, multiple datasets, all derived from scraping the SecondHandSongs [1] website, were proposed for developing VI systems [4–7]. These datasets have facilitated the development of various systems based on convolutional neural networks (CNNs) [5–12]. However, their limited sizes have restricted the feasibility of employing larger architectures, such as transformers, which are increasingly utilized in other music information retrieval (MIR) tasks [13, 14]. Additionally, existing datasets such as Da-TACOS [5] and SHS100K [4] lack comprehensive metadata, such as genre, style, and release year, which can be useful for detailed performance evaluation and sophisticated training approaches. Furthermore, they fall short in presenting sufficient challenges regarding the distribution of clique sizes, genres, styles, and track durations.

This study introduces a significantly larger and more challenging VI dataset. Rather than relying on SecondHandSongs, we use public editorial metadata from the Discogs [2] database, which has not been explored in the field previously. Discogs is collaboratively maintained by music enthusiasts and professionals who submit detailed metadata about music releases, including artist details, release information, and extensive credit descriptions. These descriptions not only list track artists and writers but also provide aliases, name variations, and artist relationships, offering a rich framework for identifying versions.

Using this metadata, we propose a methodology for identifying a large dataset of versions and mapping this dataset to various music audio collections. The resulting dataset is the largest open-source VI dataset to date. Our contributions can be summarized as follows:

1. A metadata-only dataset, Discogs-VI, containing over 1,900,000 versions of around 348,000 works.
2. A subset of this dataset, Discogs-VI-YT, containing about 493,000 versions of around 98,000 works matched to YouTube URLs of official music uploads. It contains over nine times as many works and over four times as many versions as other datasets.
3. A larger and more challenging test set that contains other publicly available test sets.
4. A pre-trained baseline model, Discogs-VINet.

The dataset [3], together with the tools for its creation, the extracted audio features, and the model trained on this data [4], are publicly available online.

Attribution: R. O. Araz, X. Serra, and D. Bogdanov, "Discogs-VI: A Musical Version Identification Dataset Based on Public Editorial Metadata", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.,* San Francisco, United States, 2024.

[1] https://secondhandsongs.com/
[2] https://www.discogs.com/
[3] https://mtg.github.io/discogs-vi-dataset/
[4] https://github.com/raraz15/Discogs-VINet

Dataset	Source	Cliques	Versions	MCS	ACS	mCS	A-URL	m-URL	OV	Content
covers80 [15]	private	80	160	2	2	2	-	-	-	Full audio, title, album, artist
YouTubeCovers [16]	YouTube	50	350	7	7	7	-	-	✗	Features (full track)
Da-TACOS [5]	SHS	1,000	13,000	13	13	13	1.0	1.0	✗	Features (full track), metadata
CoversDataset [6]	SHS	26,905	110,794	24	4	3	1.0	1.0	✗	Features (first 3 min)
SHS-100K [4]	SHS	9,999	116,353	387	12	8	1.0	1.0	✗	Title, artist
Discogs-VI-YT	Discogs	98,785	493,049	658	5	2	1.5	1.0	✓	Rich metadata, features (full track)
Discogs-VI	Discogs	348,796	1,911,611	1,837	6	2	-	-	-	Rich metadata

Table 1. Overview of publicly-available VI datasets. Da-TACOS refers to the benchmark subset, for which the 2,000 noise works are not reported as they do not form cliques. SHS refers to the SecondHandSongs website. MCS: maximum clique size; ACS: average clique size; mCS: median clique size; A-URL: average YouTube URLs per version; m-URL: median YouTube URLs per version; OV: use of official YouTube videos only. "-" denotes that the property is not applicable.

2. IDENTIFYING VERSIONS ON DISCOGS

Discogs database metadata has been previously used in other MIR tasks [17–19]. In this section, we describe the proposed methodology to identify versions and cliques using its metadata. The complete Discogs data is shared as monthly data dumps under a Public Domain license, making it easy to access. In our study, we used the July 2024 data dump.

Numerous metadata fields are provided for releases, tracks, and artists, some of which are relevant for VI. We use the track title, track artists, featuring track artists, release artists, track writer artists, and release writer artists metadata. The artist metadata contains unique artist IDs and provides information regarding group memberships, artist aliases, and artist name variations, which we use extensively. In addition, we include genre, style, record label, release format, release date, master release, and release country metadata that can be potentially useful.

2.1 Version finding from metadata

We use two critical pieces of information to establish the version relationship between two tracks: the track title and the track writer artists, indicated by the "Written-By" metadata field. Specifically, we consider two tracks with the same title and a shared writer artist as versions. This is a sufficient but not necessary condition since two tracks with different names can also be versions. Nonetheless, this condition facilitates finding a significant amount of cliques and versions from the database with high precision.

The search for cliques operates on a set of tracks from the database whose track titles are normalized by applying string processing. This includes transliterating Latin characters by removing diacritics, removing leading articles, replacing "&" with "and", eliminating any text within parentheses, and removing punctuation marks. These steps aim to mitigate potential differences in metadata between different releases and eliminate mix or edit indicators enclosed in parentheses, e.g., "(Radio Edit)", thus facilitating the process of identifying cliques. Later, such differences are considered for differentiating between versions.

Using the normalized track titles, we partition the set of tracks into disjoint subsets using exact string matching. Then, we further partition these subsets by the common track writer relation to distinguish different cliques with the same title. To do so, we compile a set of writer artist IDs for every track. Given that an artist on Discogs may represent a group with several members, we extend our collection to contain all associated members and incorporate each artist's known aliases and name variations. As a result of the two-step partitioning, tracks that have the same normalized title and share a track writer are joined in the same cliques. We opted for the shared writer approach because not all writers are consistently included in credits on some releases.

Once the cliques are formed, we identify different versions by the track or release artists. In cases where track artists metadata is available, it is used; otherwise, the release artists metadata is used. If there are featuring track artists, they are also included. Therefore, a set of tracks belonging to the same clique and performed by the same set of artists is defined as a version. After identifying the versions, we discard the cliques with only one version.

In previous VI datasets, versions are not treated as sets of tracks as in our dataset. This difference arises because Discogs often lists multiple releases for essentially the same version of a track, which may vary only by the year or country of the release. Without direct access to these releases, it is impossible to confirm their differences in advance. Therefore, we treat such tracks as identical versions. Remarkably, our dataset comprehensively includes a variety of version types as systematized in [20], including live versions, remixes, and radio edits, which add valuable diversity and potential utility.

The resulting dataset, Discogs-VI, contains numerous cliques and versions. Statistics about the dataset in comparison to other datasets are provided in Table 1.

2.2 Limitations

Due to the complex processes of composing, performing, and releasing music, along with issues related to incomplete or inaccurate metadata, there are potential issues related to our approach.

Title variability: Versions can have different names, e.g. "Moon Over Naples" is the original version of both "Spanish Eyes" and "Blue Spanish Eyes". Due to having different names, our algorithm falsely places these versions into different cliques. To address this issue, complementary data from SecondHandSongs or a large language model with music history knowledge can be used.

Rule-based text matching: Even for a single language, capturing all syntactic variations with simple rules is difficult. Yet, the database contains many languages with different syntaxes. A music named-entity recognition model may help to resolve this issue.

Metadata ambiguity: "You're My Everything" is credited to "Miles Davis" in some releases while to "Miles Davis and John Coltrane", and to "The Miles Davis Quintet" in others. These credential differences often arise from practical or legal reasons associated with publishing music. However, we can not know beforehand if they are different versions using only metadata. To reduce duplicate versions, we treat them as the same version.

3. VERSION SEARCH IN YOUTUBE

Owing to its detailed metadata, Discogs-VI can be mapped to music audio catalogs or other metadata sources. For our research purposes, we use YouTube. To match the Discogs metadata of a version to the YouTube metadata of a video, we design a rule-based algorithm.

In the matching process, we only accept videos provided by an official distributor, which can be the artists themselves or third parties such as record labels. This approach is adopted because we expect the official uploads to have more accurate metadata and be more persistent on the platform over time. Consequently, our dataset is the only VI dataset containing official uploads exclusively. In addition, due to this selectivity, our algorithm demonstrates high retrieval accuracy.

Discogs provides YouTube URL annotations for some of the releases associated with versions. However, these annotations are not on the track level and they are rarely provided. For a unified approach, we instead query YouTube for all versions. The queries are created using the Discogs version metadata in the format "artist1, artist2 - track title", and if featuring artist information is available, we concatenate "(featuring artist3)". We then store the top five results for each query and apply our metadata-matching algorithm to all stored results, which allows alternative URLs for certain versions.

As a result, we successfully matched 34% of the versions of Discogs-VI to a YouTube URL. Between these matched versions, we were able to download 98% successfully, corresponding to 33% of the total versions. We then discarded the versions that were not downloaded and the cliques without at least two downloaded versions to create the Discogs-VI-YT dataset. It contains 26% of the versions and 28% of the cliques of Discogs-VI.

3.1 Metadata matching algorithm

From Discogs metadata, our algorithm utilizes the track title, track artists, or, if unavailable, the release artists, along with any featuring artists. From YouTube, it uses the video's category, uploader, artist, description, duration, and title. We process the strings similarly to the method described in Section 2.1, except that the punctuation marks and possible texts within parentheses are not deleted to identify different versions.

The algorithm initially checks if the video metadata contains the "Music" category and if the video is an official YouTube upload. We consider a video official under the following conditions: an artist or a label provided the video, which is indicated in the video description; the video uploader is an artist topic channel auto-generated by YouTube; or the Discogs artist name is the same as the video uploader's. Videos with a duration longer than 20 minutes are discarded to deal with the potential but unlikely issue of tracks sharing their titles with their albums or EPs, which could lead to full-release audio downloads.

If a video metadata passes these controls, we use the title and artist information to decide a match. If two titles are equal, we use the artist information. If the titles do not match exactly, we apply some heuristics to strip the video title from any additional information related to remastering, HD, lyrics, etc., and re-attempt the match. We then compare all possible permutations to deal with video titles in the "artist1, artist2 - track title (featuring artist3)" format, using exact string matching. This approach makes the dataset less noisy at the cost of losing potential matches.

3.2 Limitations

Since search results and the availability of YouTube videos can be affected by geolocation, re-creating the dataset may yield differences. [5] Moreover, some URLs may become unavailable in the future. [6] To mitigate this issue, we provide multiple YouTube URLs per version when possible. Therefore, even if the main URL becomes inactive, numerous versions can still be recovered from alternative URLs. Furthermore, since we only include official uploads, the probability of a video disappearing should be lower than in other datasets. These features have not been considered in previous datasets that share YouTube URLs [4, 21].

Another limitation of our methodology is that less than 8% of versions are matched to the same YouTube URLs. Analysis showed that almost all of these versions are members of the same cliques. For the cliques that exhibit this issue, we manually kept one of the duplicate versions.

4. DATASET ANALYSIS

Following the methodologies described in Section 2 and Section 3, we created the Discogs-VI and Discogs-VI-YT datasets, respectively. Table 1 reports their sizes. The large amount of detailed metadata in Discogs-VI shows great potential: combined with an industrial-scale music audio catalog, it can create new possibilities for VI system development. Moreover, Discogs-VI-YT contains more clique and version audio than all the others combined, promising to boost model performance and generalization capability.

The range of clique sizes in our dataset is unparalleled by others in the field. The presence of cliques with many

[5] We conducted YouTube queries from Barcelona, Spain.

[6] The URLs were accessed between March 2023 and July 2024.

versions is beneficial for metric learning, as it provides numerous examples within each clique [22]. The average, median, and maximum clique sizes in the dataset indicate that the distribution has a long tail, with the weight concentrated on small clique sizes. Unlike other datasets, this distribution is highly representative of real use cases.

Figure 1 reports the genre distribution of Discogs-VI-YT, demonstrating significant coverage over 13 genres. The distribution of styles, which is included in the project repository, covers 512 styles from Mambo to Tech House. Importantly, such genre metadata opens new possibilities for developing and evaluating VI systems. Previous studies have not delved into genre and style analyses, leaving their effect on performance underexplored. Given that our dataset contains relatively reliable genre and style annotations [7] such analysis is now possible [17].

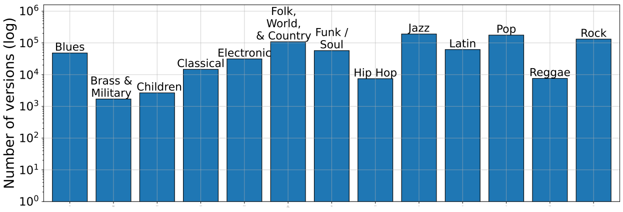

Figure 1. Discogs-VI-YT version genre distribution.

Table 2 compares the total number of artists of several VI datasets. Da-TACOS and SHS100K datasets provide only one artist per version while Discogs-VI offers multiple. For a consistent comparison, we count one artist per Discogs-VI version and do not include the group members. In addition, Da-TACOS noise works are not considered. The number of versions and artists comparisons between SHS100K and Discogs-VI-YT implies that our dataset contains more versions per artist on average.

Dataset	Artists
Da-TACOS	6,375
SHS100K	34,170
Discogs-VI-YT	67,345
Discogs-VI	239,949

Table 2. Number of track artist comparison between selected datasets. One artist per version is reported.

Figure 2 reports the audio duration distribution of Discogs-VI-YT, reflecting a comprehensive music collection. We observed that the long-duration tracks are mostly live versions and jazz or electronic music tracks, which can be notoriously long. Having long tracks increases the difficulty of training VI systems due to requiring effective time aggregation techniques or small embedding dimensions.

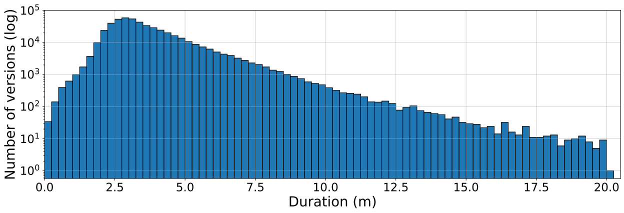

Figure 2. Discogs-VI-YT audio duration distribution.

4.1 Development and test splits

We split the Discogs-VI-YT dataset into training, validation, and test sets. To increase the compatibility with other datasets, the cliques in Discogs-VI that intersect with the Da-TACOS benchmark and SHS100K-Test sets are ensured to be part of our test set. We excluded CoversDataset from this consideration due to its lack of metadata.

To determine the intersection between our dataset and the Da-TACOS benchmark set, we conducted a thorough comparison of track titles and track writers using artist names, aliases, and name variations. We successfully identified 935 out of the 1,000 (93%) Da-TACOS cliques and 1,412 out of the 2,000 (71%) “noise” tracks. Given the detailed artist metadata we employed, it is unlikely that the unidentified works are included in our training set. Moreover, since Da-TACOS selects its “noise” tracks from those lacking alternate versions and our Discogs-VI consists exclusively of tracks with at least two versions, these tracks are also unlikely to be included in our training set. Regarding the SHS100K-Test set, we identified 1,555 out of the 1,692 cliques (90%). The union of the identified cliques from both datasets is reserved for our test set.

We aimed for a 90-10% development-test split; therefore, we sampled new cliques to add to the reserved cliques. While sampling the additional cliques, we did not exclude the SHS100K-Train set to use our dataset without restrictions. The reserved cliques from the Da-TACOS benchmark and SHS100K-Test sets had large enough sizes in our dataset. Moreover, similar to [7], we believe that having small-sized cliques in the test set simulates real use cases better. Therefore, we randomly sampled the additional cliques from sizes two to six. The remaining cliques were assigned to the development set and were further partitioned into training and validation sets following a 90-10% split. Figure 3 shows the clique size distribution of our splits, and Table 3 compares the split sizes of different datasets.

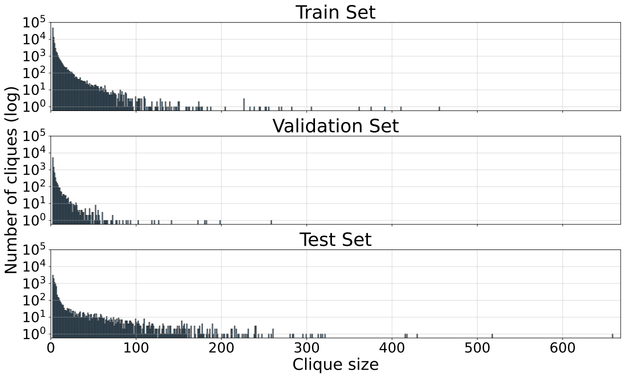

Figure 3. Discogs-VI-YT splits clique size distributions.

[7] Discogs genre and style annotations are release-level, however, they serve as a reasonable approximation for individual tracks.

Dataset	Split	Cliques	Versions	MCS	ACS	mCS
Da-TACOS	Benchmark	1,000	13,000	13	13	13
	Noise	2,000	2,000	-	-	-
SHS100K	Test	1,692	10,547	162	6	5
	Validation	1,842	10,884	17	6	6
	Train	5,324	87,091	359	16	12
Discogs-VI-YT	Test	9,878	116,197	658	12	3
	Validation	8,890	37,081	258	4	2
	Train	80,017	339,771	455	4	2

Table 3. Dataset partition sizes. MCS: maximum clique size; ACS: average clique size; mCS: median clique size

4.2 Audio representations

We computed the following audio representations commonly used in VI systems: chroma, HPCP [2], and CQT [23]. They are available under request for non-commercial scientific research purposes.

5. BASELINE MODEL

To demonstrate the utility of Discogs-VI-YT we search for a baseline model that uses computationally inexpensive input representations and is feasible for training on a consumer-grade GPU.

TPP-Net [8] and its successor CQT-Net [10] rely on the classification loss for training. Due to the large number of cliques in Discogs-VI-YT, these models are difficult to train on this dataset without modifications. ByteCover [11], ByteCover2 [12], and LyraC-Net [24] are also difficult to train as they employ the classification loss with additional losses and feature complex architectures having significantly more parameters. Additionally, the code and pre-trained weights for these three models are not publicly available. We do not consider ByteCover3 [25] and CoverHunter [26] as they do not target full-track inputs. MOVE [9] and Re-MOVE [3] are not considered due to their reliance on computationally expensive input representations. Ultimately, we selected CQT-Net, primarily due to its adaptability for use with Discogs-VI-YT.

5.1 CQT-Net

The original model is trained with the classification task, where clique IDs of the SHS100K dataset are used as class labels. A multi-length training strategy that presents the model with three different segments from each version is used to reduce possible biases toward input duration. Additionally, tempo change and spectral masking data augmentation techniques are used. During retrieval, the classification head is discarded and the remaining network is used for extracting version embeddings, whose similarity is computed with cosine similarity.

5.2 Discogs-VINet

Training CQT-Net with classification loss is challenging due to the large number of cliques in Discogs-VI-YT. Therefore, we utilize the triplet loss, similar to previous research [6, 9]. To this end, we remove the classification head from the architecture and change the affine projection layer to a linear projection with 512-dimensional outputs. Additionally, we include an L_2 normalization layer to ensure that embeddings lie on the unit hypersphere. The resulting model contains 5.2 million parameters.

At each training iteration, a mini-batch is created by randomly sampling 48 distinct cliques and two random versions per clique. With this configuration, each sample can only have one positive; hence, the positive mining strategy is equivalent to offline random sampling. For mining negatives, we use online hard-negative mining.

We extract the CQT input representations before training with CQT-Net's setting. However, we store them with 16-bit precision due to the large storage requirement of our dataset. Unlike CQT-Net's multi-length training strategy, we use fixed-length inputs where consecutive CQT frames of about 185 seconds are taken randomly. Then the features are mean downsampled with a factor of 20, following the authors. To demonstrate the benefits of using our large dataset, we do not use any data augmentation method during training, such as tempo and key modifications, spectral masking techniques, or audio degradation methods used in previous VI research.

We train Discogs-VINet for 50 epochs, which takes about 25 hours using a single Nvidia RTX2080. We use the AdamW optimizer, setting the initial learning rate to 1e-3 and adjusting via exponential decay. The triplet loss margin is set to 0.1.

During training, we use our validation set to monitor performance. Every five epochs, we simulate the VI task and save the best model in terms of mean average precision (MAP). However, we evaluate the model at the end of the training on Discogs-VI-YT, Da-TACOS, and SHS100K datasets using MAP and the mean rank of the first relevant item (MR1) metrics.

5.3 Evaluation on Discogs-VI-YT

Due to potential overlaps between the training sets of publicly available VI models and the Discogs-VI-YT test set, we could not benchmark the publicly available models. For instance, as discussed in Section 4.1, there can be shared tracks with the SHS100K-Train set. Similarly, the Da-TACOS training set, which is not publicly available, may share tracks with our test set, rendering comparisons with models trained on this dataset unreliable. Additionally, as discussed in Section 5, training numerous models on Discogs-VI-YT were not possible. We acknowledge these limitations and suggest that benchmarking models is a critical area for future research.

Despite these challenges, we present the scores obtained by Discogs-VINet. Our model obtains a MAP score of 0.443 and an MR1 score of 614.1 on the Discogs-VI-YT test set, which establishes the baseline scores on this dataset. The contrast between the MR1 and MAP values can be attributed to the realistic clique size distribution. As shown in Figure 3, the test set contains numerous cliques with size two. When a query is made with a ver-

Training data	Model	d	Da-TACOS		SHS100K-Test		SHS100K-Test**	
			MAP ↑	MR1 ↓	MAP ↑	MR1 ↓	MAP ↑	MR1 ↓
Da-TACOS	MOVE [9]	4,000	0.495	48†	✗	✗	✗	✗
	MOVE [9]	16,000	0.507	46†	✗	✗	✗	✗
	Re-MOVE [3]	256	0.524	43†	✗	✗	✗	✗
SHS100K-Train	TTP-Net [8]	300	✗	✗	0.465	72	✗	✗
	CQT-Net [10]	300	✗	✗	0.655	55	✗	✗
	ByteCover [11]	2,048	✗	✗	0.836	47	✗	✗
	ByteCover2 [12]	128	✗	✗	0.839	46	✗	✗
	ByteCover2 [12]	1,536	✗	✗	0.863	39	✗	✗
SHS100K-Train*	ByteCover [11]	2,048	0.714	23	✗	✗	✗	✗
	ByteCover2 [12]	128	0.718	23	✗	✗	✗	✗
	ByteCover2 [12]	1,536	0.791	19	✗	✗	✗	✗
SHS100K-Train**	LyraC-Net [24]	1,024	✗	✗	✗	✗	0.765	48
Private	LyraC-Net [24]	1,024	0.813	15	✗	✗	0.884	33
Discogs-VI-YT	Discogs-VINet	512	0.607	24	✗	✗	0.660	61

Table 4. Performance comparison on the Da-TACOS benchmark and SHS100K-Test sets. * denotes that the Da-TACOS benchmark set tracks were removed, ** denotes that the corresponding authors of that model downloaded the available URLs (therefore LyraC-Net [24] and Discogs-VINet are not evaluated on the same data), d denotes the embedding dimension, ✗ denotes that the result was not available, and † denotes the corrected calculations described in Section 5.4.

sion from these cliques, retrieving the only other version in high rankings contributes significantly to the MAP metric.

5.4 Evaluation on Da-TACOS and SHS100K

We tested Discogs-VINet on the Da-TACOS benchmark and SHS100K-Test sets. From the SHS100K-Test set, we could download 8,489 versions (80% of the total). As discussed in Section 4.1, we perform an extensive analysis to ensure that our training set has a minimal intersection with the evaluated sets.

The results are presented in Table 4, relying on results reported in the literature except for MOVE and Re-MOVE, for which we recomputed the results due to a metric calculation problem we discovered. In the public Da-TACOS evaluation script, "noise" works are wrongly boosting the MR1 score instead of being excluded. We corrected this issue, tested the official MOVE and Re-MOVE models, and listed the updated MR1 values.

In Table 4, Discogs-VINet outperforms both MOVE and Re-MOVE on the Da-TACOS benchmark set, which is a significant improvement given the simplicity of our input representation and lack of data augmentations. Unlike such, Discogs-VINet does not depend on pre-trained models for input representation. As a result, it exhibits significantly faster embedding extraction, similar to those reported in [12].

On the SHS100K-Test set, even though we used a slightly smaller subset due to some URLs becoming unavailable, we could not improve over other considered models, except for the TTP-Net and CQT-Net. In particular, CQT-Net, which we modified for our baseline, performed similarly. We posit that these differences may stem from the absence of data augmentation techniques in our methodology or from the classification loss possibly structuring the latent space more effectively than the triplet loss we implemented. Nonetheless, further experiments are required.

ByteCover, ByteCover2, and LyraC-Net outperform Discogs-VINet by a significant margin. This performance difference can be attributed to several factors: the combined use of classification and triplet losses, as reported in the literature [27], the advantages obtained by training larger architectures, or the absence of data augmentations in our model. However, it is important to note that independent studies have raised concerns about the reproducibility of the published results associated with the ByteCover approach [24, 28].

6. CONCLUSION

We presented a new methodology to create a VI dataset from a previously unused metadata source, Discogs. Using this metadata, we identified a large number of cliques and versions to create the Discogs-VI dataset and matched a large portion of the versions with official YouTube URLs to create its Discogs-VI-YT subset. Our datasets surpass existing datasets by far in size and provide unprecedented metadata detailing genre, style, and artist relationships.

To demonstrate the utility of Discogs-VI-YT, we trained a baseline model, Discogs-VINet, on the training set and evaluated the model performance on the test set, establishing baseline results. Additionally, we assessed Discogs-VINet's performance on the Da-TACOS benchmark and SHS100K-Test sets, where it demonstrated competitive performance. Notably, our model achieved these results without relying on any data augmentation techniques, multiple training losses, or complex architectural designs.

We leave training large models, using the metadata relations for training and evaluation, and investigating the role of data augmentations as future work.

7. ACKNOWLEDGEMENTS

This work is supported by "IA y Música: Cátedra en Inteligencia Artificial y Música" (TSI-100929-2023-1) funded by the Secretaría de Estado de Digitalización e Inteligencia Artificial and the European Union-Next Generation EU, under the program Cátedras ENIA.

8. REFERENCES

[1] J. Serrà, M. Zanin, C. Laurier, and M. Sordo, "Unsupervised Detection Of Cover Song Sets: Accuracy Improvement And Original Identification," in *Proc. of the 10th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2009.

[2] J. Serrà, E. Gómez, and P. Herrera, "Audio Cover Song Identification and Similarity: Background, Approaches, Evaluation, and Beyond," in *Advances in Music Information Retrieval*, Z. W. Raś and A. A. Wieczorkowska, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 307–332.

[3] F. Yesiler, J. Serrà, and E. Gómez, "Less is more: Faster and better music version identification with embedding distillation," in *Proc. of the 21st Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2020.

[4] X. Xu, X. Chen, and D. Yang, "Key-Invariant Convolutional Neural Network Toward Efficient Cover Song Identification," in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2018.

[5] F. Yesiler, C. Tralie, A. Correya, D. F. Silva, P. Tovstogan, E. Gómez, and X. Serra, "Da-TACOS: A Dataset for Cover Song Identification and Understanding," in *Proc. of the 20th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2019.

[6] G. Doras and G. Peeters, "Cover Detection Using Dominant Melody Embeddings," in *Proc. of the 20th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2019.

[7] ——, "A Prototypical Triplet Loss for Cover Detection," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[8] Z. Yu, X. Xu, X. Chen, and D. Yang, "Temporal Pyramid Pooling Convolutional Neural Network for Cover Song Identification," in *Proc. of the 28th Int. Joint Conf. on Artificial Intelligence*, 2019.

[9] F. Yesiler, J. Serrà, and E. Gómez, "Accurate and Scalable Version Identification Using Musically-Motivated Embeddings," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[10] Z. Yu, X. Xu, X. Chen, and D. Yang, "Learning a Representation for Cover Song Identification Using Convolutional Neural Network," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[11] X. Du, Z. Yu, B. Zhu, X. Chen, and Z. Ma, "Bytecover: Cover Song Identification Via Multi-Loss Training," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[12] X. Du, K. Chen, Z. Wang, B. Zhu, and Z. Ma, "Bytecover2: Towards Dimensionality Reduction of Latent Embedding for Efficient Cover Song Identification," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[13] T. Zeng and F. C. M. Lau, "Training audio transformers for cover song identification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2023, no. 1, p. 31, Aug. 2023.

[14] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Efficient Supervised Training of Audio Transformers for Music Representation Learning," in *Proc. of the 24th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2023.

[15] D. P. W. Ellis, "The covers80 cover song data set," 2007. [Online]. Available: https://labrosa.ee.columbia.edu/projects/coversongs/covers80/

[16] D. F. Silva and V. M. A. Souza, "Music Shapelets For Fast Cover Song Recognition," in *Proc. of the 16th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2015.

[17] D. Bogdanov, A. Porter, H. Schreiber, J. Urbano, and S. Oramas, "The Acousticbrainz Genre Dataset: Multi-Source, Multi-Level, Multi-Label, And Large-Scale," in *Proc. of the 20th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2019.

[18] D. Bogdanov and X. Serra, "Quantifying Music Trends And Facts Using Editorial Metadata From The Discogs Database," in *Proc. of the 18th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2017.

[19] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Music Representation Learning Based on Editorial Metadata from Discogs," in *Proc. of the 23rd Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2022.

[20] J. Serrà, "Identification of Versions of the Same Musical Composition by Processing Audio Descriptions," Ph.D. dissertation, Universitat Pompeu Fabra, Spain, 2011.

[21] L. A. Lanzendörfer, F. Grötschla, E. Funke, and R. Wattenhofer, "DISCO-10M: A large-scale music dataset," in *Proc. of the 37th Int. Conf. on Neural Information Processing Systems*, 2024.

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

[23] C. Schörkhuber and A. Klapuri, “Constant-Q transform toolbox for music processing,” in *Proc. of the 7th Sound and Music Computing Conf.*, 2010.

[24] S. Hu, B. Zhang, J. Lu, Y. Jiang, W. Wang, L. Kong, W. Zhao, and T. Jiang, “WideResNet with Joint Representation Learning and Data Augmentation for Cover Song Identification,” in *Proc. Interspeech*, 2022.

[25] X. Du, Z. Wang, X. Liang, H. Liang, B. Zhu, and Z. Ma, “Bytecover3: Accurate Cover Song Identification On Short Queries,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[26] F. Liu, D. Tuo, Y. Xu, and X. Han, “CoverHunter: Cover Song Identification with Refined Attention and Alignments,” in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2023.

[27] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, “Bag of Tricks and a Strong Baseline for Deep Person Re-Identification,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[28] K. O’Hanlon, E. Benetos, and S. Dixon, “Detecting Cover Songs with Pitch Class Key-Invariant Networks,” in *IEEE 31st Int. Workshop on Machine Learning for Signal Processing (MLSP)*, 2021.