PRE-TRAINING STRATEGIES USING CONTRASTIVE LEARNING AND PLAYLIST INFORMATION FOR MUSIC CLASSIFICATION AND SIMILARITY

Pablo Alonso-Jiménez[1,2], *Xavier Favory*[1], *Hadrien Foroughmand*[1], *Grigoris Bourdalas*[1], *Xavier Serra*[2], *Thomas Lidy*[1], *Dmitry Bogdanov*[2]
Utopia Music, Switzerland[1]
Music Technology Group, Universitat Pompeu Fabra, Spain[2]

ABSTRACT

In this work, we investigate an approach that relies on contrastive learning and music metadata as a weak source of supervision to train music representation models. Recent studies show that contrastive learning can be used with editorial metadata (e.g., artist or album name) to learn audio representations that are useful for different classification tasks. In this paper, we extend this idea to using playlist data as a source of music similarity information and investigate three approaches to generate anchor and positive track pairs. We evaluate these approaches by fine-tuning the pre-trained models for music multi-label classification tasks (genre, mood, and instrument tagging) and music similarity. We find that creating anchor and positive track pairs by relying on co-occurrences in playlists provides better music similarity and competitive classification results compared to choosing tracks from the same artist as in previous works. Additionally, our best pre-training approach based on playlists provides superior classification performance for most datasets.

Index Terms— music representation learning, contrastive learning, music classification, music similarity, pre-training neural networks

1. INTRODUCTION

Learning better representations is crucial to improve the quality of music classification and similarity models. Many popular approaches apply end-to-end models to learn representations while optimizing classification objectives [1, 2]. Other directions include pre-training models on editorial metadata [3, 4, 5, 6, 7], multi-modal correspondence [8], co-listening statistics [7], contrastive supervised [9, 10, 11] and self-supervised [12, 13, 14, 15, 16] objectives, music generative models [17], playlist co-occurrences [11], text [18], or combinations of them [7, 6, 17, 11]. Recently, contrastive learning has shown promising results in audio and music representation learning, especially in self-supervised fashions [14, 15], and some studies suggest that it allows learning more robust features than classification objectives [16].

Scientific evidence suggests that, in contrastive setups, it is beneficial to choose positive pairs that share information relevant for the downstream task while being diverse with respect to irrelevant characteristics [19]. However, most audio and music self-supervised contrastive methods rely on sample mixing [16], audio effects [12], or temporal crops [20] to generate the augmented versions, which intuitively have a small potential to obtain samples that are distinct enough.

Accounting for this observation, a recent study inspired by COLA [20] shows that selecting the positive pairs according to editorial metadata co-occurrences (e.g., songs from the same artist) improves the learned representations significantly [21]. In this work, we extend this method to operate with new sources of music metadata. Specifically, we focus on music consumption metadata in the form of playlists. We propose strategies to obtain positive pairs by (i) randomly sampling tracks co-occurring in playlists, (ii) constraining the positive pairs to the top co-occurrences across playlists, and (iii) using alternative track representations obtained using a Word2Vec [22] model trained on the playlist sequences as associated pair. We pre-train models based on the ResNet50 [23] and VGGish [24] architectures with playlists from the Million Playlist Dataset [25] (MPD) and then transfer the learned representations to solve music classification and similarity tasks.

The main contributions of this work are the following:

- We compare the performance of three models based on playlist data and four baselines using two different architectures in one similarity and five classification tasks.
- We propose pre-training strategies using playlist information that lead to superior performance compared to previous approaches based on editorial metadata in several music classification tasks.
- We show that some models trained with playlists achieve better similarity metrics than those based on self-supervision or editorial metadata.

The rest of this manuscript is organized as follows: Section 2 provides further motivation for the exploration of consumption metadata as a source of supervision, Section 3 describes the proposed pre-training methods, Section 4 provides details about the experimental setup, and in Section 5 we present and discuss the results. Finally, Section 6 outlines the principal conclusions of this work.

2. MOTIVATION

Self-supervised approaches enable training models with a large amount of unannotated data, which has been successful in fields such as natural language processing [26]. In practice, these approaches have a limited scope for domains where collecting unlabeled data on a large scale is difficult due to copyright limitations or simple shortage. In these cases, certain forms of weak supervision may compensate for the lack of data. For example, researchers have shown that contextual metadata can be used for representation learning of biomedical images [27] or document editorial metadata for document classification [28]. Music is also rich in metadata, which motivates using this information to train models.

Such information can be divided into *editorial metadata* used to catalog music (e.g., artist and album names, or country and year of release), and *consumption metadata* describing interactions of humans (or machines) with music (e.g., playlists, DJ setlists, radio programs, or listening histories). In this paper, we focus on the latter type. Using consumption metadata as a source of similarity ground truth has already been explored in the recommender-systems literature, enabling tasks such as music playlist continuation [25]. Also, while editorial relations are normally one- or few-to-many (e.g., album-songs), consumption is many-to-many (e.g., playlists-songs), resulting in a more dense co-occurrence space that may favor associating more heterogeneous music.[1] Furthermore, the usage of consumption metadata for music representation learning has not been as extensively investigated yet [7, 11] as the case of editorial metadata [3, 4, 5, 6, 7, 29].

3. METHOD

We investigate methods to obtain targets from music playlist datasets to pre-train models using contrastive learning.

3.1. Contrastive learning setup

Our architecture consists of a convolutional backbone $B(\cdot)$ and a projector $H(\cdot)$ that map a mel-spectrogram input $x \in \mathbb{R}^{T \times F}$ with T timestamps and F frequencies into latent representations $z \in \mathbb{R}^{D}$, and $z' \in \mathbb{R}^{D'}$ respectively. The model is trained to bring z'_x close to z'_y while pulling it apart from samples in the same batch following SimCLR [30]. After pre-training, $H(\cdot)$ is discarded, and $B(\cdot)$ is used in the downstream tasks. Our setup is depicted in Figure 1.

3.2. Pair generation algorithms

Instead of using augmentations to obtain x and y as done in SimCLR, we propose to use pairs originating from different tracks by exploiting playlist information. The number of possible pairs of elements that co-occur in a playlist of size n corresponds to the number of combinations without repetition $\binom{n}{2}$. This produces many pairs when considering millions of playlists, making the exhaustive usage of the pairs difficult to scale with this contrastive learning approach. Because of this, we propose algorithms that rely on heuristics to create audio pairs that utilize a wide range of tracks while preventing track repetitions, as well as an embedding learning-based technique to create the target pairs.

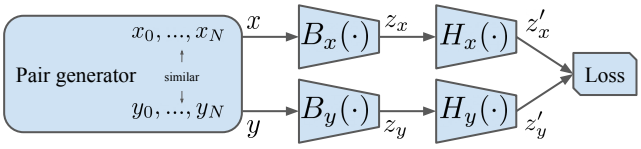

Fig. 1. Illustration of our pre-training pipeline. The features x and y from the associated pairs are input to the model $B(\cdot)$ and projector $H(\cdot)$. $B(\cdot)$ and $H(\cdot)$ are optimized using a contrastive loss. $B(\cdot) = B_x(\cdot) = B_y(\cdot)$ and $H(\cdot) = H_x(\cdot) = H_y(\cdot)$ in all the cases except for *Word2Vec representation*.

Considering a dataset of *playlists* $P = \{p_0, ..., p_N\}$, and *tracks* $S = \{s_0, ..., s_M\}$, we propose the following strategies:

- *Co-Occurrence*. This approach randomly generates pairs by producing combinations using the available tracks in each playlist p_i and with each track appearing in only one pair. We iterate randomly through P generating $\lfloor \frac{|p_i|}{2} \rfloor$ pairs per playlist and discarding the associated tracks from the set of available tracks. This algorithm is executed at the beginning of each training epoch.
- *Top Co-Occurrence*. This algorithm counts the number of co-occurrences of the tracks in all the playlists. For each track we randomly select its associated pair among its top-10 most co-occurring tracks while ensuring that every track appears only in one pair. To do so, at each epoch, we initialize a set of available tracks $A = S$. We randomly iterate through A and for a given track s_j we select one of the top co-occurring tracks s_k and discard s_j and s_k from A.
- *Word2Vec representation*. This is a multi-modal approach in which, for a given track, we align the projection of its audio representation z'_x to the projection of its *Word2Vec* embedding z'_y [22]. We train a *Word2Vec* model by considering playlists as sentences and track ids as words. We rely on the Continuous Bag of Words approach with a context window that includes the entire playlist [2] and a learning rate of 0.02 for 20 epochs.[3] In this case $B_y(\cdot)$ is the frozen pre-trained *Word2Vec* model, z_y is a *Word2Vec* embedding, and $H_y(\cdot)$ is a different projector from $H_x(\cdot)$ featuring the same hyper-parameters and dimensions.

4. EXPERIMENTS

Our experiments are divided into two steps. First, we pre-train the proposed models in a contrastive setup. These models

[1] For example, our dataset has an average number of tracks per artist and playlist of 7.2 and 66.3, respectively. On average, a track appears on 29.3 playlists and belongs to 1.28 artists.

[2] We also tested a W2V sensitive to the track positions in the playlists by using smaller window sizes. However, this degraded the performance.

[3] We use the Gensim implementation `https://radimrehurek.com/gensim/models/word2vec.html`

Model	Pairs per epoch	Pair generation algorithm
SimCLR	1,779,072	-
Artist CO	1,014,528*	*Co-Occurrence*
Playlist CO	826,368*	*Co-Occurrence*
Playlist TCO	731,520*	*Top Co-Occurrence*
Playlist W2V	1,779,072	*Word2Vec representation*

Table 1. Number of pairs per epoch and pair generation algorithms. *indicates that these are different pairs on each epoch.

are then fine-tuned and evaluated in the downstream music classification or directly evaluated in a music similarity task.

4.1. Pre-training

We pre-train a number of models following the different pair generation strategies. *SimCLR* is a baseline where x and y are alternative views of the same audio patch mixed with random patches from the batch scaled with a gain factor sampled from a $\beta(5, 2)$ distribution similar to previous work [16]. *Artist CO* is another baseline that applies the *Co-Occurrence* strategy to the artist names (i.e., considering the set of tracks by each artist as a playlist) without preventing track repetitions.

Playlist CO, *Playlist TCO*, and *Playlist W2V* use the *Co-Occurrence*, *Top Co-Occurrence*, and *Word2Vec representation* strategies respectively to generate the x/y pairs using the playlist information. Table 1 shows the number of pairs per epoch and model. In *SimCLR* and *Playlist W2V*, the number of pairs corresponds to the number of tracks in the dataset since these methods do not associate different tracks. In *Playlist CO* and *Playlist TCO*, we constrain to a single track occurrence per epoch, which results in fewer pairs per epoch. We train the models for a fixed number of 50 epochs. This makes the pair generation algorithms execute the same number of times, which leads to a different number of batch optimization steps for each model.

We pre-train all our models using the Million Playlist Dataset (*MPD*) [25] matched to our in-house music collection, which resulted in 1,779,072 tracks and 999,219 playlists. $H(\cdot)$ has a single hidden layer with 128 units and a ReLu activation and $D' = 128$. We use the NT-Xent loss [30] with a fixed τ value of 0.1 using a batch size of 384 pairs, and the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is increased linearly from 0 to $1e$-4 for the first 5,000 steps and then decreased following a cosine decay until the models complete 50 epochs similar to [16]. We train the models on 96-band, 256-timestamp ($\sim$3 seconds) mel-spectrogram patches randomly selected at each iteration from the 30-seconds excerpts available for each track.

4.2. Music classification

Our first evaluation consists of solving multi-label music classification tasks by fine-tuning the pre-trained models. We keep the pre-trained $B(\cdot)$ and replace $H(\cdot)$ by an MLP with the same hidden layer configuration and output dimensions matching the number of classes followed by a Sigmoid activation. We optimize $B(\cdot)$ and the new $H(\cdot)$ using Adam ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) and cross-entropy loss with an L2 regularization term of $1e$-5 for a maximum of 50 epochs. We use a cyclical triangular scheduler that varies the learning rate from $1e$-5 to $1e$-4 [31]. The weights are selected from the epoch with the highest Average Precision on the validation set. We apply early stopping after ten epochs without any improvement on this metric. In training, we use the same random patch selection approach as in pre-training. During inference, we average the activations from non-overlapping patches. We use 30 seconds of audio from the center of the track in validation, and the full duration available in testing.

We use the Genre, Instrument, and Mood subsets of the MTG-Jamendo Dataset [32], containing 55,215, 25,135, and 18,4856 full tracks, and 87, 40, and 56 classes, respectively. We consider the MagnaTagATune (MTAT) dataset [33], with 25,860 30-seconds excerpts, and its top-50 tags using the 12:1:3 partition [34]. Additionally, we consider an in-house genre dataset containing 87,542 2-minutes excerpts and 72 classes, referred to as Genre Internal. Our goal is to assess if our pre-training approaches are still beneficial when a bigger and arguably more curated collection is available.

4.3. Music similarity

For the music similarity evaluation, we use the dim-sim dataset consisting of a collection of music similarity triplets produced by human raters [35]. Each triplet was annotated by 5 to 12 people, and the official clean version of the dataset contains 879 triplets with a high inter-annotator agreement. We extract representations z for the clean subset of dim-sim using the pre-trained models without fine-tuning. Following the common evaluation approach [35], we measure the cosine distance between anchor/positive, and anchor/negative, and consider the triplet prediction correct if the latter is larger. We report the prediction accuracy and the average difference between anchor/negative and anchor/positive distances.

4.4. Architectures

We consider two standard backbone architectures:

- **VGGish** [24]. This is a variant of the VGG [36] architecture popular in the audio domain. It has 128 output dimensions. We consider the original model weights obtained from a classification task in a proprietary dataset as a baseline.[4] When pre-training the architecture with our data, we use our 3-second 96-bands mel-spectrogram patches and modify the kernel of the first pooling layer from 2×2 to 4×4 to keep the number of dimensions after the convolutional layers close to the one in the original model.
- **ResNet50** [23]. We use the standard ResNet50 model considering its good performance in audio and music applications [16]. We reduce the output of the last dense layer with global max- and mean-pooling and concatenate the resulting vectors, leading to an output embedding of 4,096 dimensions.

[4] https://github.com/tensorflow/models/tree/master/research/audioset/vggish

Dataset	Genre		Instrument		Mood		MTAT		Genre Internal	
	AP	ROC	AP	ROC	AP	ROC	AP	ROC	AP	ROC
					VGGish					
VGGish FT	15.8±0.3	84.9±0.5	18.1±0.6	74.2±1.2	12.1±0.9	72.7±0.8	44.4±0.6	90.6±0.1	-	-
From Scratch	13.4±0.2	82.6±0.3	15.5±0.4	72.1±0.5	9.3±0.2	70.6±0.4	40.2±0.7	88.9±0.2	54.3±0.3	96.7±0.0
SimCLR	15.2±0.3	83.6±0.4	16.4±0.3	72.1±0.5	10.7±0.2	70.1±0.2	41.1±0.6	88.9±0.2	61.7±0.1	97.6±0.0
Artist CO	17.3±0.1	85.6±0.1	20.4±0.4	76.7±0.1	13.9±0.2	74.5±0.5	46.2±0.1	91.1±0.1	68.8±0.2	98.3±0.1
Playlist CO	17.0±0.1	85.4±0.2	20.2±0.4	76.1±0.3	13.3±0.8	73.8±0.8	45.9±0.2	90.9±0.0	67.7±0.7	98.2±0.1
Playlist TCO	17.5±0.1	84.9±0.4	20.5±0.3	76.3±0.9	13.8±0.3	73.7±0.7	45.8±0.3	91.0±0.1	70.0±0.4	98.4±0.0
Playlist W2V	17.3±0.2	85.5±0.3	19.8±0.7	75.3±0.2	13.5±0.1	72.8±0.7	45.3±0.6	90.9±0.2	69.8±0.1	98.4±0.0
					Resnet50					
From Scratch	14.4±0.2	82.9±0.1	15.6±0.5	71.2±0.6	8.9±0.0	69.2±0.4	40.7±0.3	88.8±0.1	63.3±0.1	97.7±0.1
SimCLR	16.3±0.2	84.7±0.3	17.4±0.1	73.3±0.7	12.1±0.3	73.0±0.3	43.4±0.5	90.1±0.2	67.4±0.3	98.2±0.0
Artist CO	**19.0±0.1**	85.0±0.1	21.1±0.4	76.5±0.9	14.9±0.3	74.8±0.7	47.0±0.3	**91.5±0.2**	73.4±0.2	98.6±0.1
Playlist CO	18.7±0.7	**85.7±0.4**	**21.2±0.7**	76.7±0.9	14.8±0.5	74.2±0.4	46.8±0.2	91.4±0.0	73.4±0.1	98.6±0.0
Playlist TCO	18.9±0.2	85.1±0.3	20.4±0.7	75.4±1.5	14.3±0.4	73.7±0.7	**47.0±0.2**	91.3±0.2	72.8±0.2	98.6±0.0
Playlist W2V	**19.0±0.1**	85.4±0.3	20.7±0.4	**77.1±0.4**	**15.0±0.1**	**75.1±0.4**	46.7±0.4	91.2±0.2	**74.1±0.2**	**98.7±0.1**

Table 2. Metrics in the music classification datasets expressed in macro ROC-AUC and Average Precision. For each architecture, we present the baselines on top and the proposed models below. Metrics statistically equivalent or higher than *Artist CO* according to a one-sided t-test ($p\text{-}value = 0.005$) are marked in light grey. The highest metric per dataset is marked in bold.

Model	VGGish	Resnet50	VGGish	Resnet50
	Accuracy		Average difference	
SimCLR	0.699	0.672	0.007	0.009
Artist CO	0.819	0.838	0.043	0.039
Playlist CO	**0.852**	**0.845**	**0.077**	**0.064**
Playlist TCO	0.793	0.813	0.052	0.046
Playlist W2V	0.831	0.818	0.067	0.041

Table 3. Music similarity accuracy and the average difference between anchor/negative and anchor/positive.

5. RESULTS AND DISCUSSION

Table 2 shows the macro ROC-AUC and Average Precision[5] metrics for all the datasets and models as the average ± the standard deviation of three runs. Our baselines consist of fine-tuning the original VGGish [24] (*VGGish FT*), randomly initialized models (*From Scratch*), *SimCLR*, and *Artist CO*.

Firstly, we note that the contrastive approaches based on artist and playlist metadata always achieve better performance than the *VGGish FT*, *From Scratch*, and *SimCLR* baselines, which aligns with previous works indicating the benefits of metadata-based supervision [3, 4, 5, 21]. The models based on playlist information achieve equivalent or superior performance to those based on *Artist CO* on most datasets and metrics, and *Playlist W2V* with the ResNet50 architecture achieves the best performance in at least one metric for the Genre, Instrument, Mood, and Genre Internal datasets.

Table 3 contains the results of the music similarity evaluation. We observe that models based on metadata show a stronger correlation with human similarity perception than the baseline SimCLR approach. While *Playlist CO* achieves the best metrics with both architectures, *Playlist TCO* and *Playlist W2V* did not improve the performance as in the classification tasks. We hypothesize that *Top Co-Occurrence* and *Word2Vec representation* reduce the diversity of the positive pairs, which may augment the discriminative capabilities of the latent space at the cost of becoming weaker for similarity.

Finally, these results may depend on the nature and sparsity of the available playlists. In our study, we relied on MPD, which contains a curated subset of Spotify playlists filtered by quality and enriched with additional tracks. The playlists were created by US users only between 2010 and 2017 and are not expected to be representative of the overall distribution of Spotify playlists. However, MPD represents a small fraction of more than 4 billion playlists on Spotify, which motivates further research on playlist-based pre-training.

6. CONCLUSIONS

In this work, we show that employing contrastive learning for pre-training neural networks with playlist information is valuable for music classification. While previous works focused on editorial metadata, such as the artist name, we found that superior performance can be achieved with consumption metadata consisting of playlist information by relying on track representations obtained from a Word2Vec model trained on the playlist sequences. Also, the representations learned using simple playlist co-occurrences perform significantly better than an unsupervised approach (SimCLR) or than using artist co-occurrences for music similarity. Future work includes validating our approaches with more sources of consumption metadata (e.g., radio programs or listening histories) considering learning them in multi-task scenarios.

7. ACKNOWLEDGEMENTS

This research was partially funded by Musical AI - PID2019-111403GB-I00/AEI/10.13039/501100011033 of the Spanish Ministerio de Ciencia, Innovación y Universidades.

[5] Average Precision is also referred to as the area under the precision-recall curve (PR-AUC) in the literature.

8. REFERENCES

[1] Sander Dieleman and Benjamin Schrauwen, "End-to-end learning for music audio," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[2] Keunwoo Choi, Gyorgy Fazekas, and Mark Sandler, "Automatic tagging using deep convolutional neural networks," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2016.

[3] J. Park, Jongpil Lee, Jung-Woo Ha, and Juhan Nam, "Representation learning of music using artist labels," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2018.

[4] Jaehun Kim, Minz Won, Xavier Serra, and Cynthia C. S. Liem, "Transfer learning of artist group factors to musical genre classification," *Intl. World Wide Web Conf.*, 2018.

[5] Jongpil Lee, Jiyoung Park, and Juhan Nam, "Representation learning of music using artist, album, and track information," in *Intl. Conf. on Machine Learning (ICML), Machine Learning for Music Discovery Workshop*, 2019.

[6] Jaehun Kim, Julián Urbano, Cynthia C. S. Liem, and Alan Hanjalic, "One deep music representation to rule them all? a comparative analysis of different representation learning strategies," *Neural Computing and Applications*, 2020.

[7] Qingqing Huang, Aren Jansen, Li Zhang, Daniel PW Ellis, Rif A Saurous, and John Anderson, "Large-scale weakly-supervised content embeddings for music recommendation and tagging," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[8] Jason Cramer, Ho-Hsiang Wu, Justin Salamon, and Juan Pablo Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[9] Xavier Favory, Konstantinos Drossos, Tuomas Virtanen, and X. Serra, "COALA: co-aligned autoencoders for learning semantically enriched audio representations," in *Workshop on Self-supervised learning in Audio and Speech, Intl. Conf. on Machine Learning (ICML)*, 2020.

[10] Xavier Favory, Konstantinos Drossos, Tuomas Virtanen, and Xavier Serra, "Learning contextual tag embeddings for cross-modal alignment of audio and tags," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[11] Andres Ferraro, Xavier Favory, Konstantinos Drossos, Yuntae Kim, and Dmitry Bogdanov, "Enriched music representations with multiple cross-modal contrastive learning," *Signal Processing Letters*, 2021.

[12] Janne Spijkervet and John Ashley Burgoyne, "Contrastive learning of musical representations," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2021.

[13] Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino, "Byol for audio: Self-supervised learning for general-purpose audio representation," in *2021 Intl. Joint Conf. on Neural Networks (IJCNN)*, 2021.

[14] Dong Yao, Zhou Zhao, Shengyu Zhang, Jieming Zhu, Yudong Zhu, Rui Zhang, and Xiuqiang He, "Contrastive learning with positive-negative frame mask for music representation," in *ACM Web Conf.*, 2022.

[15] Hang Zhao, Chen Zhang, Bilei Zhu, Zejun Ma, and Kejun Zhang, "S3t: Self-supervised pre-training with swin transformer for music classification," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[16] Luyu Wang, Pauline Luc, Yan Wu, Adria Recasens, Lucas Smaira, Andrew Brock, Andrew Jaegle, Jean-Baptiste Alayrac, Sander Dieleman, Joao Carreira, et al., "Towards learning universal audio representations," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[17] Rodrigo Castellon, Chris Donahue, and Percy Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2021.

[18] Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas, "Learning music audio representations via weak language supervision," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[19] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola, "What makes for good views for contrastive learning?," *Advances in Neural Information Processing Systems*, 2020.

[20] Aaqib Saeed, David Grangier, and Neil Zeghidour, "Contrastive learning of general-purpose audio representations," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[21] Pablo Alonso-Jiménez, Xavier Serra, and Bogdanov Dmitry, "Music representation learning based on editorial metadata from discogs," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2022.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. of the Conf. on computer vision and pattern recognition (CVPR)*, 2016.

[24] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al., "CNN architectures for large-scale audio classification," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[25] Ching-Wei Chen, Paul Lamere, Markus Schedl, and Hamed Zamani, "RecSys challenge 2018: Automatic music playlist continuation," in *ACM Conf. on Recommender Systems*, 2018.

[26] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang, "Pre-trained models for natural language processing: A survey," *Science China Technological Sciences*, 2020.

[27] Stephan Spiegel, Imtiaz Hossain, Christopher Ball, and Xian Zhang, "Metadata-guided visual representation learning for biomedical images," *BioRxiv*, 2019.

[28] Natraj Raman, Armineh Nourbakhsh, Sameena Shah, and Manuela Veloso, "Domain-agnostic document representation learning using latent topics and metadata," in *Intl. FLAIRS Conf.*, 2021.

[29] Pablo Alonso-Jiménez, Dmitry Bogdanov, and Xavier Serra, "Deep embeddings with essentia models," *Late-Braeking/Demo, Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2020.

[30] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *Intl. Conf. on Machine Learning (ICML)*, 2020.

[31] Leslie N Smith, "Cyclical learning rates for training neural networks," in *Winter Conf. on Applications of Computer Vision (WACV)*, 2017.

[32] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra, "The MTG-Jamendo dataset for automatic music tagging," in *Machine Learning for Music Discovery Workshop, Intl. Conf. on Machine Learning (ICML)*, 2019.

[33] Edith Law, Kris West, Michael I. Mandel, Mert Bay, and J. Stephen Downie, "Evaluation of algorithms using games: The case of music tagging," in *Intl. Society for Music Information Retrieval Conf., (ISMIR)*, 2009.

[34] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Conf. of the Intl. Society for Music Information Retrieval (ISMIR)*, 2014.

[35] Jongpil Lee, Nicholas J. Bryan, Justin Salamon, Zeyu Jin, and Juhan Nam, "Disentangled multidimensional metric learning for music similarity," in *Proc. of the Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[36] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.