LARGE-SCALE WEAKLY-SUPERVISED CONTENT EMBEDDINGS FOR MUSIC RECOMMENDATION AND TAGGING

Qingqing Huang, Aren Jansen, Li Zhang, Daniel P. W. Ellis, Rif A. Saurous, John Anderson

Google Research, Mountain View, CA, and New York, NY USA

{qqhuang,arenjansen,liqzhang,dpwe,rif,janders}@google.com

ABSTRACT

We explore content-based representation learning strategies tailored for large-scale, uncurated music collections that afford only weak supervision through unstructured natural language metadata and co-listen statistics. At the core is a hybrid training scheme that uses classification and metric learning losses to incorporate both metadata-derived text labels and aggregate co-listen supervisory signals into a single convolutional model. The resulting joint text and audio content embedding defines a similarity metric and supports prediction of semantic text labels using a vocabulary of unprecedented granularity, which we refine using a novel word-sense disambiguation procedure. As input to simple classifier architectures, our representation achieves state-of-the-art performance on two music tagging benchmarks.

***Index Terms*—** Music information retrieval, music content embedding, word sense disambiguation, joint audio and text models

1. INTRODUCTION

Transfer learning using large-scale pretrained embedding models has had great success in multiple domains, including natural language processing, computer vision, and audio processing. Our goal is to extend this trend to cross-domain music modeling by constructing a joint embedding model of the audio and natural language terms that captures fine-grained semantic similarity and supports downstream music understanding and retrieval tasks. Our training dataset is a large, uncurated collection of over 10 million music videos from the internet, including content ranging from produced recordings to live amateur performances. Each recording is accompanied by a rich set of unnormalized metadata, including titles and descriptions, from which we derive a large vocabulary of over 100,000 text labels, some explicitly music-related and some not. Compared to the previous research, which involves only small vocabularies of hundreds of tags [1], our dataset includes a preponderance of natural language labels that enables associating finer-grain meanings to the content.

As in other uncurated data settings, our main objective is to exploit any available supervisory signals to elicit a useful content representation. We have to overcome multiple challenges in achieving this objective. First, the text labels are highly noisy since they are generated from free-form natural language titles and descriptions. Secondly, the natural language terms can be highly ambiguous and require contextual information to resolve their meanings.

To safeguard against noise in the text labels, we use the collection's accompanying co-listen statistics, which are known to be highly indicative of musical similarity [2] and thus provide a useful additional learning signal. We consider triplet loss network training as a content-based co-listen graph embedding technique that implies a music similarity metric reflecting underlying listener preferences. Unlike representations learned with collaborative filtering (CF), which embed the graph structure directly, our audio-based parametric methods inherently extend to new content. The co-listen statistics can also be used to disambiguate natural language terms. We apply CF techniques to embed and cluster the recordings, and then contextualize the text labels by associating them with different clusters. We demonstrate that splitting the vocabulary into different senses is effective in associating more precise meaning to the audio content.

We evaluate the proposed method for a range of network architectures. First, we show that our learned content embeddings define a similarity metric that can be directly used to predict co-listen graph edges on a heldout set. Second, we demonstrate that our models define a feature representation that propel simple fully-connected classifiers to state-of-the-art performance on two music tagging benchmarks: MagnaTagATune [3] and AudioSet [4].

2. RELATED WORK

Transfer learning in the music domain has seen substantial attention over the past several years, with various efforts to learn general-purpose music representations [5, 6, 7, 8, 9]. The most closely related to ours is Dieleman et al. [5], who perform unsupervised training of a convolutional deep belief network using the Million Song Database (MSD) [10], and subsequently fine-tune it for artist/genre recognition and key detection tasks. In a follow-up work, van den Oord et al. [6] trained a convolutional embedding network by regression of CF embeddings derived from co-listen data. Finally, van den Oord et al. [8] consider the more typical supervised pre-training of a representation using tags from MSD, and demonstrate utility on downstream tasks like MagnaTagATune. This provides a strongly supervised learning baseline in our experiments. More recently, music tagging models that use more advanced techniques have been evaluated in the literature. These include raw waveform CNN [11], AlexNet CNNs [12], persistent-topology CNNs [13], Convolutional Recurrent Neural Networks (CRNN) [14], sample-level CNNs [15], ResNets [16], and Squeeze-and-Excitation Networks [16].

The idea of using co-listen statistics to learn an objective measure of musical similarity has been circulating for quite some time [2]. However, only recently has it been connected to deep learning approaches in the CF embedding regression approach [6], which can be done more computationally efficiently compared to our approach. However, its accuracy is limited by the quality of CF embeddings, especially in under-sampled regions of the graph where CF embeddings lose the fine granularity. Past work has also considered triplet loss for music similarity metric learning supervised on relative human judgments [17]. Recently, Siamese networks were used for music representation learning [18], using same/different-artist song pairs as supervision. Our method is the first attempt to use co-listen statistics to train an embedding network directly from acoustic inputs deep metric learning.

3. PROPOSED APPROACH

Our training framework assumes a large collection of music audio accompanied by heterogeneous supervision types: noisy text labels and a co-listen graph. We optimize the single embedding network according to a hybrid objective, consisting of both classification and distance metric learning losses. We also introduce a novel technique to address the high level of word-sense ambiguity.

3.1. Dataset

We perform our experiments using a collection of 10.5 million soundtracks of public internet videos, each containing a majority of music content as determined by an automatic music detector. We use three types of data from the collection: (i) the audio recording; (ii) textual metadata, including its title and description; and (iii) aggregate co-listen statistics between recording pairs. Each audio recording is downsampled to 16 kHz and clipped to a maximum of 10 minutes, resulting in a total of over 800,000 hours of audio. In terms of number of audio recordings, the music collection is 10 times bigger than MSD. The scale of our data provides an opportunity for learning a rich semantic music representation, but it comes at the cost of reduced supervision quality that requires tailored techniques to accommodate.

We apply standard tokenization to the text data to obtain a set of n-grams and create the label vocabulary by choosing the most frequent 100K terms spanning multiple languages. We do not apply any filtering to the terms since it is difficult *a priori* to know which terms are important. For example, if we apply frequent word filtering (even with TF-IDF re-weighting), we may lose popular yet useful music related terms such as "happy", "fast". To address privacy concern, we sanitize co-listen statistics as follows: removing users with too few views before computing co-listen counts and dropping counts below certain threshold in the co-listen statistics. In addition, we keep at most 250 neighbors for each recording to form the co-listen graph. Such sparsification also helps improve model quality as it prevents from being biased by higher degree vertices or by accidental co-listen.

3.2. Hybrid optimization scheme

Our dataset $\mathcal{X}$ consists of a set of N music recordings. Each recording $X_i \in \mathcal{X}$ is comprised of a sequence $X_i = x_1 x_2 \dots x_{T_i}$ of spectro-temporal context windows of the form $x_t \in \mathbb{R}^{F \times T}$, where T is the window length in frames and F is the number of frequency channels. Weak supervision is provided in the following forms:

- A set of label data $\mathcal{L} = \{L_1, \dots, L_N\}$, where each $L_i \in \{0,1\}^{|\mathcal{V}|}$ is the binary-valued target vector over the label vocabulary $\mathcal{V}$ for each recording $X_i \in \mathcal{X}$.
- A sparse, unweighted, undirected co-listen graph $G = (V, E)$, where there is exactly one vertex $v(X_i) \in V$ for each recording $X_i \in \mathcal{X}$, and an edge $e(v(X_i), v(X_j)) \in E$ iff the recording pair (X_i, X_j) were both listened to by at least τ visitors.

Our goal is to learn a map $f : \mathbb{R}^{F \times T} \to \mathbb{R}^d$ that embeds spectro-temporal context windows from recordings into a d-dimensional vector space. While distance metric learning losses apply directly to the output of such an embedding network, incorporating the weak text label information requires an additional classification layer $g : \mathbb{R}^d \to [0,1]^{|\mathcal{V}|}$ that maps the output of f to a set of posterior estimates for the tags in $\mathcal{V}$. Finally, at test time, we construct an embedding $S \in \mathbb{R}^d$ for the whole recording $X = x_1 \dots x_T$ by the mean embedding of the constituent context windows, given by $S = (1/T)\sum_{t=1}^{T} f(x_t)$. Using the above notation, we define two loss function: the triplet loss $L_{\text{triplet}}(\mathcal{X}, G, f)$ for estimating how the model fits the co-listen graph; and the cross entropy loss $L_{\text{CE}}(\mathcal{X}, f, g)$ for estimating how the model fits the text labels.

3.2.1. Co-listen prediction objective

While our co-listen graph is defined at the recording level, we extend the relations to all constituent spectro-temporal windows they contain. Thus, to construct each training triplet, we sample (i) an "anchor" recording $X_a \in \mathcal{X}$; (ii) a "positive" recording X_p from $\{X | (v(X_a), v(X)) \in E\}$, i.e. the set of co-listen neighbor recordings of X_a; (iii) a difficult "negative" recording X_n from $\{X | (v(X_p), v(X)) \in E \text{ and } (v(X_a), v(X)) \notin E\}$, i.e. the set of co-listen neighbors of X_p that are not connected to X_a; and finally (iv) a triplet (x_a, x_p, x_n) by randomly sampling context windows x_a, x_p, and x_n from X_a, X_p, and X_n, respectively. Given the dataset $\mathcal{X}$ and accompanying co-listen graph G, we can generate a virtually limitless triplet stream, which we partition into batches of size B of the form $\mathcal{B} = \{(a_i, p_i, n_i)\}_{i=1}^{B}$, where each $a_i, p_i, n_i \in \mathbb{R}^{F \times T}$. The triplet loss for each batch is then defined by

$$L_{\text{triplet}}(\mathcal{B}, f) = \sum_{i=1}^{B} \left[\|f(a_i) - f(p_i)\|_2^2 - \|f(a_i) - f(n_i)\|_2^2 + \delta \right]_+ , \tag{1}$$

where $\|\cdot\|$ denotes ℓ_2-norm, $[\cdot]_+$ denotes hinge loss, and δ is a nonnegative margin hyperparameter. Despite having purposefully selected negatives according to the co-listen graph, it remains critical to perform the within-batch semi-hard negative mining procedure [19]. This involves the reassignment of triplet negatives to anchor-positive pairs to make more difficult triplets, choosing the closest negative to each anchor that is still further away than the positive.

3.2.2. Text label prediction objective

Despite the fact that our data is highly multi-labeled, our experimentation clearly showed that using a softmax classification layer significantly outperforms a layer of independent logistics when predicting the noisy text labels. Data inspection indicated a correlation between number of labels and the label noise. Using a softmax loss implies an ℓ_1 normalization to target vectors, which usefully downweights such label noise. We again extend the weak text label information from the recording level to all spectro-temporal context windows contained in each recording. We transform the dataset $\mathcal{X}$ and accompanying text label set $\mathcal{L}$ into a large set of multi-labeled context frames that we again partition into batches of size B of the form $\mathcal{B} = \{(x_i, \lambda_i)\}_{i=1}^{B}$, where $x_i \in \mathbb{R}^{F \times T}$ is a context window from some recording $X_j \in \mathcal{X}$ and $\lambda_i = L_j / \|L_j\|_1$ is the ℓ_1-normalized target vector computed from the original target vector $L_j \in \mathcal{L}$ corresponding to X_j. The softmax cross-entropy loss for a batch is defined as

$$L_{\text{CE}}(\mathcal{B}, f, g) = -\sum_{i=1}^{B} \lambda_i \cdot \log g(f(x_i)). \tag{2}$$

Here, log is applied element-wise and $g(u) = \exp(u \cdot h_j + b_j) / \sum_{k=1}^{|\mathcal{V}|} \exp(u \cdot h_k + b_k)$, where $h_j \in \mathbb{R}^d$ and $b_j \in \mathbb{R}$ are the embeddings and bias of the label $j \in \mathcal{V}$, respectively. We observe that it is necessary to include an ℓ_2 normalization in the embedding function f directly so that its outputs are guaranteed to have unit length. If not, the embedding norm may blow up dramatically as training proceeds, reducing ultimate classifier layer performance.

Such an ℓ_2 normalization is also traditionally included in the triplet loss distance function. Finally, the ℓ_2-normalized rows of the final classifier layer g define embeddings for each tag in $\mathcal{V}$.

3.2.3. *Curriculum training*

We apply a curriculum training procedure that first optimizes the the triplet loss objective, followed by the the cross-entropy loss. We choose this order because our primary interest is in creating music content embeddings that capture fine-grained semantic concepts, and text labels provide this connection in a human-interpretable fashion. However, pretraining with co-listen graph triplets encourages embedding space structure that reflects a generic notion of musical similarity and user preference. The alternative of training against a weighted sum of both losses requires a more complicated adjustment of the adaptive optimizer (Adam) for different parts of the network independently, since otherwise a single learning rate will govern the embedding network (which is affected by both losses) and the classifier layer (which is affected by only cross-entropy). With our curriculum training schedule, we simply wait for the triplet loss to level off and switch to cross-entropy optimization. At our data scale, overfitting is not a concern.

3.3. Text label disambiguation

Natural language terms can be highly ambiguous, which can have completely distinct meanings in different contexts. If we train our model directly on these labels, it would cause the ambiguous label to move to the "centroid" of the multiple recordings which contain that label. This would make the embedding of such terms represent the mixture of distinct meanings, or heavily biased by the most popular meaning, and cause confusion to the audio model to try to fit to it. Empirically, we would observe lower prediction performance for such terms. To solve this problem, we need to identify the context of the text labels, which can be inferred from the listening pattern on the "host" video. Hence we apply the weighted matrix factorization method to the co-listen graph and cluster the recordings using the K-Means algorithm. The clusters then serve as the context information for disambiguating text labels.

For each recording, we attach its cluster id to all of its text labels. In this way, each term may is split into multiple "atomic" terms, corresponding to its meaning in different clusters. We need to carefully choose the number of clusters: if too large, we may have too few recordings associated with a particular label; if too small, we may still have too much ambiguity in the text labels. A good heuristic is to select the number of clusters such that high AUC is achieved with respect to the split labels. Using this, we indeed observe that our disambiguation does associate finer meanings to the text labels while not adversely affecting the meanings of non-ambiguous terms.

4. EXPERIMENTS

We process each recording into log mel spectrograms of $F = 64$ mel bins using standard short-time Fourier analysis (0.025 s Hanning window, 0.010 s step size). The input to each one of our embedding models is a 3-second context window, producing $T = 300$ by $F = 64$ spectrogram patch inputs to each 2D convolutional neural network. We specify three convolutional embedding network architectures that provide a range of options for the complexity/quality tradeoff.

The first model specializes the relatively heavyweight ResNet-18 architecture [20] (11.2M parameters, 686M multiplies) to take our modified 2D spectrogram input, increasing the layer 1 kernel/stride to 7x5/2x1, and the subsequent max-pool kernel/stride to 5x3/4x2. Second, we modify the more efficient MobileNet architecture [21] (670k parameters, 26.1M multiplies) to also accept 300x64 inputs, changing the first conv layer's strides to 2x1 and adding an additional max-pool after the third conv layer with 4x2 kernel and 2x1 stride. The third network is a simple 3-layer 2D convolutional network with interspersed max-pooling operations, which we label SmallCNN (537k parameters and 22M multiplies). The architecture is specified by the chain [conv(7x5,2x1), max(4x2, 2x1), conv(7x5,2x1), max(4x2, 2x1), conv(7x5,2x1), max(4x2, 2x1)], where the two arguments are kernel and stride, respectively. After the final time-frequency pooling operation, we append a 128-unit fully connected layer, followed by ℓ_2-normaliztion to define the output embedding. This output is optimized directly by triplet loss (Eq. 1. To compute the cross-entropy (Eq. 2) we append a softmax classifier layer of size $|\mathcal{V}| \approx$100K, which corresponds to the non-disambiguated text labels. We then freeze the trained ResNet-18 network and retrain a softmax classifier of size 1M corresponding to the cluster id prefixed labels.

The dataset is split 90/10 into a train and test recording set. A small portion of the test set is used for validation, and a disjoint subset is used to construct evaluation examples for co-listen link prediction and text label prediction. The networks are trained with Adam optimizer using 132-example batches for both losses. We include batch norm on all convolutional layers and use a learning rate of 1e-4. The first triplet loss training stage is run until the loss begins to level off (approximately 10M steps) followed by cross-entropy minimization for another 10M steps. The triplet margin gap parameter was $\delta = 0.1$ in all cases.

Table 1. Text label prediction performance in AUC-ROC.

	Label-only			Hybrid
Model	Small CNN	MobileNet	ResNet18	ResNet18
AUC	0.788	0.835	0.855	0.859

Table 2. Link prediction performance, measured in average precision (AP) and AUC-ROC. WMF trains on evaluation data by necessity.

Model	**AP**	**AUC**
Random	0.0008	0.500
ResNet-18 CF Regression [6]	0.055	0.916
SmallCNN triplet-only	0.035	0.888
MobileNet triplet-only	0.057	0.925
ResNet-18 triplet-only	0.079	0.946
ResNet-18 hybrid	**0.107**	**0.956**
WMF (upper bound)	0.428	0.974

4.1. Text label prediction

First we consider the task of predicting our noisy text labels on a held out set, directly evaluating the model output after cross-entropy training. We use a sample of 10,000 recordings drawn from the test portion of our dataset for evaluation, restricting eval to text labels with at least 10 occurrences in this set (approximately 2,000 labels of the 100,000 label vocabulary). We average per-context window scores across each recording and compute recording-level prediction performance relative to the ground truth labels. Table 1 shows performance for several models in terms of unweighted AUC-ROC. We find that with label-only training, the largest ResNet architecture outperforms the smaller models. However, despite having a comparable size and computation cost to SmallCNN, the MobileNet comes significantly closer to ResNet performance, making it a natural choice for resource-constrained settings. Finally, we find that hybrid training provides an additional improvement over label training alone.

Table 3. Comparison of top tags to cluster id prefixed seed tags. Tags are ranked by cosine similarity on embeddings defined by softmax layer weights.

46_bebop	5_bebop	24_devotional	38_devotional
hard bop	cowboy bebop	ganesha	wilder
jimmy garrison	cowboy	radha	devotional tour
hank jones	for sale	mahadeva	fletcher
coltrane	ost anime music	monks	violator

Table 4. AUC-ROC of MagnaTagATune top-50 tag prediction.

	Train	**Classifier**	
Embedding	**Set**	**linear**	**FC-1x512**
SmallCNN hybrid fixed	full	0.903	0.911
MobileNet hybrid fixed	full	0.912	0.916
ResNet-18 hybrid fixed	full	0.910	**0.919**
ResNet-18 label-only fixed	full	0.911	0.916
ResNet-18 triplet-only fixed	full	0.888	0.902
ResNet-18 hybrid warm-start	full	0.906	**0.920**
SmallCNN hybrid fixed	10%	0.888	0.894
MobileNet hybrid fixed	10%	0.900	0.902
ResNet-18 hybrid fixed	10%	0.896	0.900
Baselines (all full-train)			
ResNet-18 cold-start	0.889	Transfer Learning [8]	0.880
Waveform CNN [11]	0.882	AlexNet [12]	0.901
SampleCNN [15]	0.906	SqueezeNet [16]	0.911

Table 5. AudioSet genre performance in AUC-ROC.

Embedding	**Train Set**	**7-genre**	**25-genre**
MobileNet hybrid fixed	full	0.930	0.915
ResNet-18 hybrid fixed	full	**0.930**	**0.916**
MobileNet hybrid fixed	10%	0.929	0.912
ResNet-18 hybrid fixed	10%	0.928	0.913
Resnetish-50 [4, 23]	full	0.914	0.901

4.2. Co-listen link prediction

Next we consider the utility of the similarity measure defined by our embedding models to characterize listener preference. To measure this, we randomly sampled two disjoint sets of 1,250 co-listen recording pairs not used in training. For each set, we computed the cosine similarities between recording-level average embeddings for all 2500-choose-2 pairs. We rank pairs by this cosine similarity, and measure the retrieval performance of the 1,250 co-listen pairs in terms of average precision and AUC-ROC. In Table 2, we again see that our largest ResNet-18 architecture significantly outperforms smaller networks, but MobileNet again bends the cost-performance curve to come in as a clear second place. Interestingly, even though the triplet training objective is closely linked to this evaluation methodology, fine-tuning with text labels still provides substantial gain. Note that weighted matrix factorization (WMF) [22] directly embeds the co-listen graph including the evaluation set, so it provides a performance upper bound; however, the performance gap for our proposed models are much smaller than would be expected from past studies. Specifically, the WMF performance recovery of our audio models—25% for AP and 98% for AUC-ROC—dwarfs that of a similar past evaluation, which recovered 2.5% and 80% of the WMF AP and AUC-ROC, respectively (see Table 3 of [6]). On our evaluation, our methods nearly double the AP of our own ResNet-18 implementation of the CF regression baseline in [6].

4.3. Examples of label disambiguation

We provide examples to demonstrate the effect of using the co-listen structure for label meaning disambiguation. We pick some terms that are present in multiple clusters and compute their nearest neighbor labels to show that different meanings of the same label are indeed captured. In Table 3, we show the results of two terms “bebop” and “devotional”, which both can refer to different concepts or entities depending on the context. In the table, we prepend the cluster id to the term, indicating the different discovered senses. For “bebop”, it sucessfully differentiated the two prevalent contexts of Jazz music and Anime soundtracks; for “devotional”, it identified the Indian sub-genre of devotional music, as well as the “Devotional Tour” by the British band Depeche Mode (in the Pop music cluster). In all the cases, the global model only captures the most common meaning.

4.4. MagnaTagATune and AudioSet tagging benchmarks

We evaluate our models as general purpose audio feature extractors for downstream tagging tasks. We begin with MagnaTagATune [3] and consider the well-exercised top-50 tag set. We use standard train/validation/test partitions, tuning hyperparameters on validation and reporting class-balanced AUC-ROC on the test set. We consider two classifier architectures on top of our 128-dimensional embeddings: (i) an independent per-class logistic regression layer, and (ii) a single-hidden layer perceptron with 512 hidden units and an independent logistic output layer. We also consider the effect of using only 10% of the training set to probe how much our embeddings support the tagging task on their own.

In Table 4, we observe that the fixed ResNet-18 embeddings trained with hybrid loss achieve the best (and state-of-the-art) performance when coupled to a fine-tuned single hidden layer MLP. However, the efficient MobileNet architecture is again a close second. Fine-tuning the embedding network provides little boost, indicating the pretrained network is already providing a generally useful representation for downstream tasks. Our cold-start ResNet-18 baseline employs the same architecture as our other ResNet-18 fixed embedding networks but achieves far lower performance, indicating that architecture alone does not account for our high performance relative to baselines. Finally, we find that even with 10% of the training data, we still outperform several baseline systems, indicating our pretrained embeddings are truly carrying most of the weight for this task.

We also consider the genre prediction task on AudioSet dataset [4], for both a 7-way genre task defined in [24], and a harder 25-way task including all top-level genre categories. Our baseline is the Resnetish-50 classifier from [23] trained on the entirety of AudioSet (all 527 classes). Table 5 shows class-balanced AUC-ROC performance for both tag sets. In all cases, we are using fixed embeddings and training a set of linear models (independent logistic regression for each class) on top. We again exceed baseline performance using our fixed hybrid-optimized embeddings, even when only using 10% of the AudioSet training data.

5. CONCLUSION

In this work, we explored a collection of weakly-supervised representation learning strategies for content-based music recommendation and tagging, and demonstrated that data scale and diversity can overcome label noise to produce robust content embedding models that achieve state-of-the-art performance on well-studied benchmarks.

6. REFERENCES

[1] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson, "Fma: A dataset for music analysis," *arXiv preprint arXiv:1612.01840*, 2016.

[2] Adam Berenzweig, Beth Logan, Daniel PW Ellis, and Brian Whitman, "A large-scale evaluation of acoustic and subjective music-similarity measures," *Computer Music Journal*, vol. 28, no. 2, pp. 63–76, 2004.

[3] Edith Law and Luis Von Ahn, "Input-agreement: a new mechanism for collecting data using human computation games," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 1197–1206.

[4] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "AudioSet: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[5] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen, "Audio-based music classification with a pretrained convolutional network," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[6] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen, "Deep content-based music recommendation," in *Advances in neural information processing systems*, 2013, pp. 2643–2651.

[7] Philippe Hamel, Matthew EP Davies, Kazuyoshi Yoshii, and Masataka Goto, "Transfer learning in mir: Sharing learned latent representations for music audio classification and similarity," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2013.

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

[9] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho, "Transfer learning for music classification and regression tasks," *arXiv preprint arXiv:1703.09179*, 2017.

[10] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere, "The million song dataset," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[11] Sander Dieleman and Benjamin Schrauwen, "End-to-end learning for music audio," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6964–6968.

[12] Umut Güçlü, Jordy Thielen, Michael Hanke, and Marcel Van Gerven, "Brains on beats," in *Advances in Neural Information Processing Systems*, 2016, pp. 2101–2109.

[13] Jen-Yu Liu, Shyh-Kang Jeng, and Yi-Hsuan Yang, "Applying topological persistence in convolutional neural network for music audio signals," *arXiv preprint arXiv:1608.07373*, 2016.

[14] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2392–2396.

[15] Jongpil Lee and Juhan Nam, "Multi-level and multi-scale feature aggregation using sample-level deep convolutional neural networks for music classification," *arXiv preprint arXiv:1706.06810*, 2017.

[16] Taejun Kim, Jongpil Lee, and Juhan Nam, "Sample-level cnn architectures for music auto-tagging using raw waveforms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 366–370.

[17] Brian McFee and Gert RG Lanckriet, "Heterogeneous embedding for subjective artist similarity.," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2009, pp. 513–518.

[18] Jiyoung Park, Jongpil Lee, Jangyeon Park, Jung-Woo Ha, and Juhan Nam, "Representation learning of music using artist labels," *arXiv preprint arXiv:1710.06648*, 2017.

[19] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 770–778.

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[22] Yifan Hu, Yehuda Koren, and Chris Volinsky, "Collaborative filtering for implicit feedback datasets.," in *ICDM*, 2008, vol. 8, pp. 263–272.

[23] 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 131–135.

[24] Hareesh Bahuleyan, "Music genre classification using machine learning techniques," *arXiv preprint arXiv:1804.01149*, 2018.