TRANSFER LEARNING BY SUPERVISED PRE-TRAINING FOR AUDIO-BASED MUSIC CLASSIFICATION

Aäron van den Oord, Sander Dieleman, Benjamin Schrauwen
Electronics and Information Systems department, Ghent University
{aaron.vandenoord, sander.dieleman, benjamin.schrauwen}@ugent.be

ABSTRACT

Very few large-scale music research datasets are publicly available. There is an increasing need for such datasets, because the shift from physical to digital distribution in the music industry has given the listener access to a large body of music, which needs to be cataloged efficiently and be easily browsable. Additionally, deep learning and feature learning techniques are becoming increasingly popular for music information retrieval applications, and they typically require large amounts of training data to work well. In this paper, we propose to exploit an available large-scale music dataset, the Million Song Dataset (MSD), for classification tasks on other datasets, by reusing models trained on the MSD for feature extraction. This transfer learning approach, which we refer to as *supervised pre-training*, was previously shown to be very effective for computer vision problems. We show that features learned from MSD audio fragments in a supervised manner, using tag labels and user listening data, consistently outperform features learned in an unsupervised manner in this setting, provided that the learned feature extractor is of limited complexity. We evaluate our approach on the GTZAN, 1517-Artists, Unique and Magnatagatune datasets.

1. INTRODUCTION

With the exception of the Million Song Dataset (MSD) [3], public large-scale music datasets that are suitable for research are hard to come by. Among other reasons, this is because unwieldy file sizes and copyright regulations complicate the distribution of large collections of music data. This is unfortunate, because some recent developments have created an increased need for such datasets.

On the one hand, content-based music information retrieval (MIR) is finding more applications in the music industry, in a large part due to the shift from physical to digital distribution. Nowadays, online music stores and streaming services make a large body of music readily available to the listener, and content-based MIR can facilitate cataloging and browsing these music collections, for example by automatically tagging songs with relevant terms, or by creating personalized recommendations for the user. To develop and evaluate such applications, large music datasets are needed.

On the other hand, the recent rise in popularity of feature learning and deep learning techniques in the domains of computer vision, speech recognition and natural language processing has caught the attention of MIR researchers, who have adopted them as well [13]. Large amounts of training data are typically required for a feature learning approach to work well.

Although the initial draw of deep learning was the ability to incorporate large amounts of unlabeled data into the models using an unsupervised learning stage called *unsupervised pre-training* [1], modern industrial applications of deep learning typically rely on purely supervised learning instead. This means that large amounts of labeled data are required, and labels are usually quite costly to obtain.

Given the scarcity of large-scale music datasets, it makes sense to try and leverage whatever data is available, even if it is not immediately usable for the task we are trying to perform. We can use a *transfer learning* approach to achieve this: given a target task to be performed on a small dataset, we can train a model for a different, but related task on another dataset, and then use the learned knowledge to obtain a better model for the target task.

In image classification, impressive results have recently been attained on various datasets by reusing deep convolutional neural networks trained on a large-scale classification problem: ImageNet classification. The ImageNet dataset contains roughly 1.2 million images, divided into 1,000 categories [5]. The trained network can be used to extract features from a new dataset, by computing the activations of the topmost hidden layer and using them as features. Two recently released software packages, *OverFeat* and *DeCAF*, provide the parameters of a number of pre-trained networks, which can be used to extract the corresponding features [7, 20]. This approach has been shown to be very competitive for various computer vision tasks, sometimes surpassing the state of the art [18, 26].

Inspired by this approach, we propose to train feature extractors on the MSD for two large-scale audio-based song classification tasks, and leverage them to perform other classification tasks on different datasets. We show that this approach to transfer learning, which we will refer to as *supervised pre-training* following Girshick et al. [9],

consistently improves results on the tasks we evaluated.

The rest of this paper is structured as follows: in Section 2, we give an overview of the datasets we used for training and evaluation. In Section 3 we describe our proposed approach and briefly discuss how it relates to transfer learning. Our experiments and results are described in Section 4. Finally, we draw conclusions and point out some directions for future work in Section 5.

2. DATASETS

The **Million Song Dataset** [3] is a collection of metadata and audio features for one million songs. Although raw audio data is not provided, we were able to obtain 30 second preview clips for almost all songs from 7digital.com. A number of other datasets that are linked to the MSD are also available. These include the **Taste Profile Subset** [15], which contains listening data from 1 million users for a subset of about 380,000 songs in the form of play counts, and the **last.fm dataset**, which provides tags for about 500,000 songs. We will use the combination of these three datasets to define two *source tasks*: user listening preference prediction and tag prediction from audio.

We will evaluate four *target tasks* on different datasets:

- genre classification on the **GTZAN dataset** [22], which contains 1,000 audio clips, divided into 10 genres.
- genre classification on the **Unique** dataset [21], which contains 3,115 audio clips, divided into 14 genres.
- genre classification on the **1517-artists** dataset [21], which contains 3,180 full songs, divided into 19 genres.
- tag prediction on the **Magnatagatune** dataset [14], which contains 25,863 audio clips, annotated with 188 tags.

3. PROPOSED APPROACH

3.1 Overview

There are many ways to transfer learned knowledge between tasks. Pan and Yang [17] give a comprehensive overview of the transfer learning framework, and of the relevant literature. In their taxonomy, our proposed supervised pre-training approach is a form of *inductive transfer learning* with *feature representation transfer*: target labels are available for both the source and target tasks, and the feature representation learned on the source task is reused for the target task.

In the context of MIR, transfer learning has been explored by embedding audio features and labels from various datasets into a shared latent space with linear transformations [10]. The same shared embedding approach has previously been applied to MIR tasks in a multi-task learning setting [24]. We refer to these papers for a discussion of some other work in this area of research.

For supervised pre-training, it is essential to have a source task that requires a very rich feature representation, so as to ensure that the information content of this representation is likely to be useful for other tasks. For computer vision problems, ImageNet classification is one such task, since it involves a wide range of categories. In this paper, we will evaluate two source tasks using the MSD: tag prediction and user listening preference prediction from audio. The goal of tag prediction is to automatically determine which of a large set of tags are associated with a given song. User listening preference prediction involves predicting whether users have listened to a given song or not.

Both tasks differ from typical classification tasks in a number of ways:

- Tag prediction is a *multi-label classification* task: each song can be associated with multiple tags, so the classes are not disjoint. The same goes for user listening preference prediction, where we attempt to predict for each user whether they have listened to a song. The listening preferences of different users are not disjoint either, and one song is typically listened to by multiple users.
- There are large numbers of tags and users; orders of magnitude larger than the 1,000 categories of ImageNet.
- The data is weakly labeled: if a song is not associated with a particular tag, the tag may still be applicable to the song. In the same way, if a user has not listened to a song, they may still enjoy it (i.e. it would be a good recommendation). In other words, some positive labels are missing.
- The labels are redundant: a lot of tags are correlated, or have the same meaning. For example, songs tagged with *disco* are more likely to also be tagged with *80's*. The same goes for users: many of them have similar listening preferences.
- The labels are very sparse: most tags only apply to a small subset of songs, and most users have only listened to a small subset of songs.

We will tackle some of the problems created by these differences by first performing dimensionality reduction in the label space using *weighted matrix factorization* (WMF, see Section 3.2), and then training models to predict the reduced label representations instead.

We will first use the spherical K-means algorithm (see Section 3.3) to learn low-level features from audio spectrograms, and use them as input for the supervised models that we will train to perform the source tasks. Feature learning using K-means is very fast compared to other unsupervised feature learning methods, and yields competitive results. It has recently gained popularity for content-based MIR applications [6, 19, 25].

In summary, our workflow will be as follows: we will first learn low-level features from audio spectrograms, and apply dimensionality reduction to the target labels. We will train supervised models to predict the reduced label representations from the extracted low-level audio features. These models can then be used to perform the source tasks. Next, we will use the trained models to extract higher-level features from other datasets, and use those features to train shallow classifiers for different but related target tasks. We will compare the higher-level features obtained from different model architectures and different source tasks by

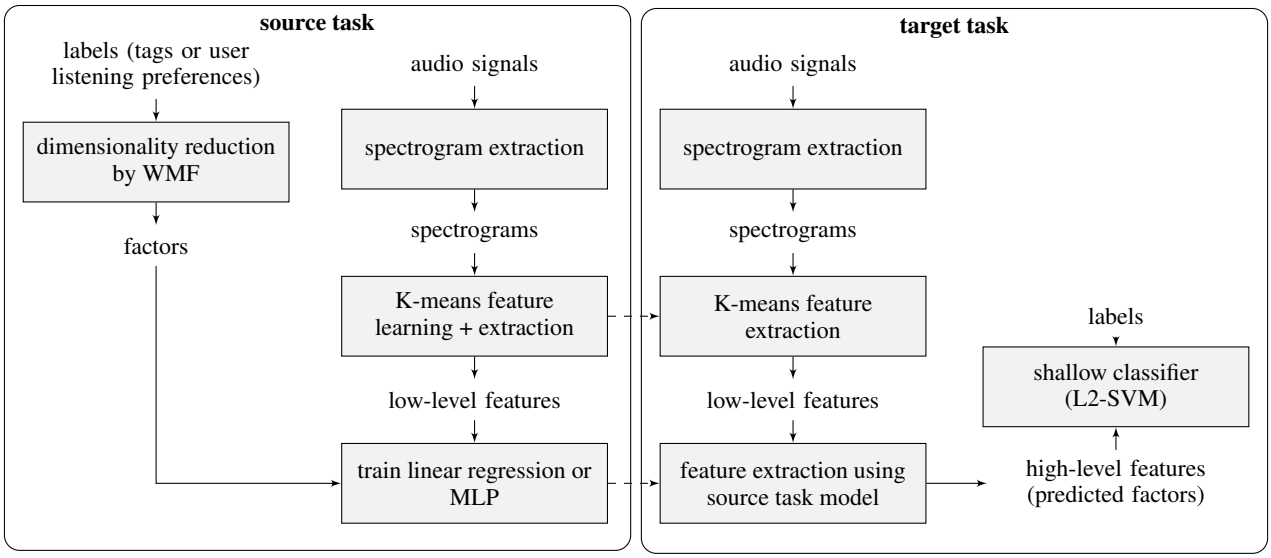


Figure 1: Schematic overview of the workflow we will use for our supervised pre-training approach. Dashed arrows indicate transfer of the learned feature extractors from the source task to the target task.

evaluating their performance on these target tasks. This workflow is visualized in Figure 1. The key learning steps are detailed in the following subsections.

3.2 Dimensionality reduction in the label space

To deal with large numbers of overlapping labels, we first consider the matrix of labels for all examples, and perform weighted matrix factorization (WMF) on it [12]. Given a binary $m \times n$-matrix A (m examples and n labels), WMF will find an $m \times f$-matrix U and an $n \times f$-matrix V, so that $A \approx UV^T$. The hyperparameter f controls the rank of the resulting approximation. This approximation is found by optimizing the following weighted objective function:

$$J(U, V) = C \circ (A - UV^T)^2 + \lambda(||U||_F^2 + ||V||_F^2),$$

where C is a $m \times n$ *confidence matrix*, $\circ$ represents elementwise multiplication, the squaring is elementwise as well, and λ is a regularization parameter. If the confidence values in C are chosen to be 1 for all zeroes in A, an efficient alternating least squares (ALS) method exists to optimize $J(U, V)$, provided that A is sparse. For details, we refer to Hu et al. [12].

After optimization, each row of U can be interpreted as a reduced representation of the m labels associated with the corresponding example, which captures the latent factors that affect its classification. We can then train a model to predict these f factors instead, which is much easier than predicting m labels directly (typically $f \ll m$). We have previously used a similar approach to do content-based music recommendation with a convolutional neural network [23]. In that paper, we showed that these factors capture a lot of relevant information and can also be used for tag prediction. We use the same settings and hyperparameter values for the WMF algorithm in this work.

Our choice for WMF over other dimensionality reduction methods, such as PCA, is motivated by the particular structure of the label space described earlier. WMF allows for the sparsity and redundancy of the labels to be exploited, and we can take into account that the data is weakly labeled by choosing C so that positive signals are weighed more than negative signals.

The original label matrix for the tag prediction task has 173,203 columns, since we included all tags from the last.fm dataset that occur more than once. The matrix for the user listening preference prediction task has 1,129,318 columns, corresponding to all users in the Taste Profile Subset. By applying WMF, we obtain reduced representations with 400 factors for both tasks. These factors will be treated as ground truth target values in the supervised learning phase.

3.3 Unsupervised learning of low-level features

We learn a low-level feature representation from spectrograms in an unsupervised manner, to use as input for the supervised pre-training stage. First, we extract log-scaled mel-spectrograms from single channel audio signals, with a window size of 1024 samples and a hop size of 512. Conversion to the mel scale reduces the number of frequency components to 128. We then use the spherical K-means algorithm (as suggested by Coates et al. [4]) to learn 2048 bases from randomly sampled PCA-whitened windows of 4 consecutive spectrogram frames. This is similar to the feature learning approach proposed by Dieleman et al. [6].

To extract features, we divide the spectrograms into overlapping windows of 4 frames, and compute the dot product of each base with each PCA-whitened window. We then aggregate the feature values across time by computing the maximal value for each base across groups of consecutive windows corresponding to about 2 seconds of audio. Finally, we take the mean of these values across the entire audio clip to arrive at a 2048-dimensional feature representation for each example. This two-stage temporal pooling approach turns out to work well in practice.

3.4 Supervised learning of high-level features

For both source tasks, we train three different model architectures to predict the reduced label representations from the low-level audio features: a linear regression model, a multi-layer perceptron (MLP) with a hidden layer with 1000 rectified linear units (ReLUs) [16], and an MLP with two such hidden layers. The MLPs are trained using stochastic gradient descent (SGD) to mimize the mean squared error (MSE) of the predictions, and dropout regularization [11]. The training procedure was implemented using Theano [2].

We trained all these models on a subset of the MSD, consisting of 373,855 tracks for which we were able to obtain audio samples, and for which listening data is available in the Taste Profile Subset. We used 308,443 tracks for training, 18,684 for validation and 46,728 for testing. For the tag prediction task, the set of tracks was further reduced to 253,588 tracks, including only those for which tag data is available in the last.fm dataset. For this task, we used 209,218 tracks for training, 12,763 for validation and 31,607 for testing.

The trained models can be used to extract high-level features simply by computing predictions for the reduced label representations and using those as features, yielding feature vectors with 400 values. For the MLPs, we can alternatively compute the activations of the topmost hidden layer, yielding feature vectors with 1000 values instead. The latter approach is closer to the original interpretation of supervised pre-training as described in Section 1, but since the trained models attempt to predict latent factor representations, the former approach is viable as well. We will compare both.

To evaluate the models on the source tasks, we compute the predicted factors U' and obtain predictions for each class by computing $A' = U'V^T$. This matrix can then be used to compute performance metrics.

3.5 Evaluation of the features for target tasks

To evaluate the high-level features for the target tasks outlined in Section 2, we train linear L2-norm support vector machines (L2-SVMs) for all tasks with liblinear [8], using the features as input. Although using more powerful classifiers could probably improve our results, the use of a shallow, linear classifier helps to assess the quality of the input features.

4. EXPERIMENTS AND RESULTS

4.1 Source tasks

To assess whether the models trained for the source tasks are able to make sensible predictions, we evaluate them by computing the normalized mean squared error (NMSE) [1] of the latent factor predictions, as well as the area under the ROC curve (AUC) and the mean average precision (mAP) of the class predictions [2]. They are reported in Table 1. Note that the latter two metrics are computed on a separate test set, but the former is computed on the validation set that we also used to optimize the hyperparameters for the dimensionality reduction of the labels. This is because the ground truth latent factors, which are necessary to compute the NMSE, are not available for the test set.

User listening preference prediction			
Model	**NMSE**	**AUC**	**mAP**
Linear regression	0.986	0.750	0.0076
MLP (1 hidden layer)	0.971	0.760	0.0149
MLP (2 hidden layers)	0.961	0.746	0.0186

Tag prediction			
Model	**NMSE**	**AUC**	**mAP**
Linear regression	0.965	0.823	0.0099
MLP (1 hidden layer)	0.939	0.841	0.0179
MLP (2 hidden layers)	0.924	0.837	0.0179

Table 1: Results for the source tasks. For all three models, we report the normalized mean squared error (NMSE) on the validation set, and the area under the ROC curve (AUC) and the mean average precision (mAP) on a separate test set.

It is clear that using a more complex model (i.e. an MLP) results in better predictions of the latent factors in the least-squares sense, as indicated by the lower NMSE values. However, when using the AUC metric, this does not always seem to translate into better performance for the task at hand: MLPs with only a single hidden layer perform best for both tasks in this respect. The mAP metric seems to follow the NMSE on the validation set more closely.

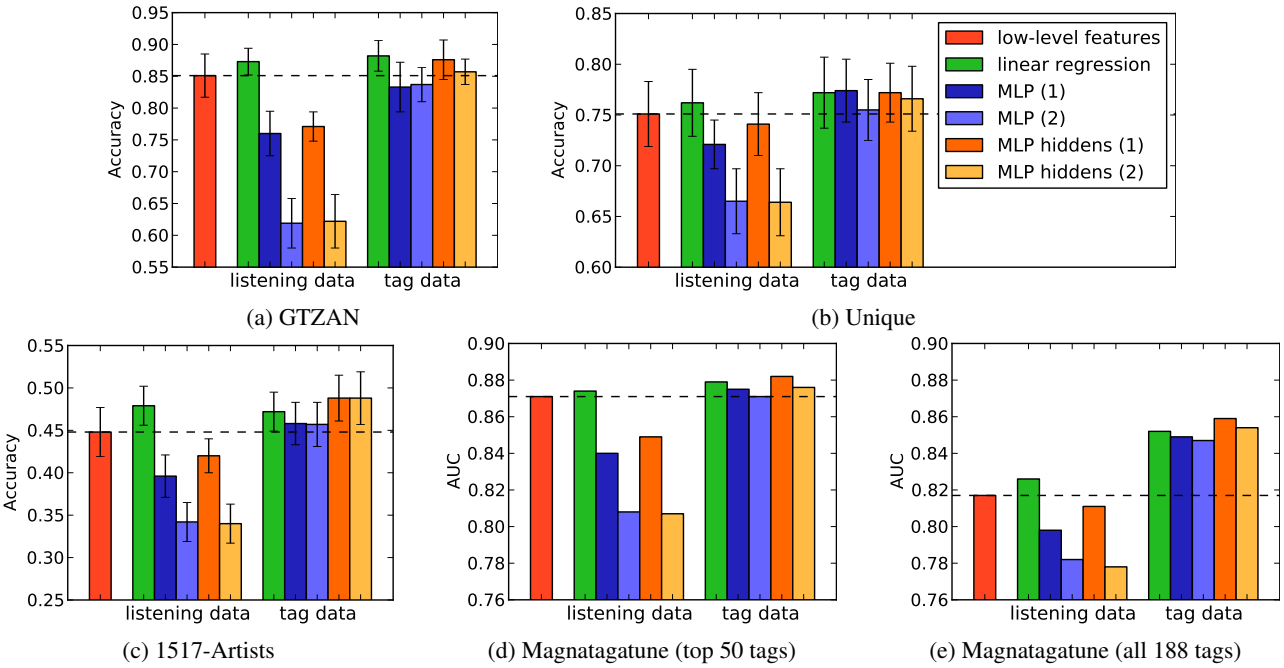
Although the NMSE values are relatively high, the class prediction metrics indicate that the predicted factors still yield acceptable results for the source tasks. In our preliminary experiments we also observed that using fewer factors tends to result in lower NMSE values. In other words, as we add more factors, they become less predictable. This implies that the most important latent factors extracted from the labels are also the most predictable from audio.

4.2 Target tasks

We report the L2-SVM classification performance of the different feature sets across all target tasks in Figure 2. For the GTZAN, Unique and 1517-Artists datasets, we report the average cross-validation classification accuracy across 10 folds. Error bars indicate the standard deviations across folds. We optimize the SVM regularization parameter using nested cross-validation with 5 folds. Magnatagatune comes divided into 16 parts; we use the first 11 for training and the next 2 for validation. After hyperparameter optimization, we retrain the SVMs on the first 13 parts, and the last 3 are used for testing. We report the AUC aver-

[1] The NMSE is the MSE divided by the variance of the target values across the dataset.

[2] The class predictions are obtained by multiplying the factor predictions with the matrix V^T, as explained in the previous section.



(a) GTZAN (b) Unique

(c) 1517-Artists (d) Magnatagatune (top 50 tags) (e) Magnatagatune (all 188 tags)

Figure 2: Target task performance of the different feature sets. The dashed line represents the performance of the low-level features. From left to right, the five bars in the bar groups represent high-level features extracted with linear regression, an MLP with 1 hidden layer, an MLP with 2 hidden layers, the hidden layer of a 1-layer MLP, and the topmost hidden layer of a 2-layer MLP respectively. Error bars for the first three classification tasks indicate the standard deviation across cross-validation folds. For Magnatagatune, no error bars are given because no cross-validation was performed.

aged across tags for the 50 most frequently occuring tags (Figure 2d), and for all 188 tags (Figure 2e).

The single bar on the left of each graph shows the performance achieved when training an L2-SVM directly on the low-level features learned using spherical K-means. The two groups of five bars show the performance of the high-level features trained in a supervised manner for the user listening preference prediction task and the tag prediction task respectively.

Across all tasks, using the high-level features results in improved performance over the low-level features. This effect is especially pronounced for Magnatagatune, when predicting all 188 tags from the high-level features learned on the tag prediction source task. This makes sense, as some of the Magnatagatune tags are quite rare, and features learned on this closely related source task must contain at least some relevant information for these tags.

Comparing the performance of different source task models for user listening preference prediction, model complexity seems to play a big role. Across all datasets, features learned with linear regression perform much better than MLPs, despite the fact that the MLPs perform better for the source task. Clearly the MLPs are able to achieve a better fit for the source task, but in the context of transfer learning, this is actually a form of overfitting, as the features generalize less well to the target tasks – they are too specialized for the source task. This effect is not observed when the source task is tag prediction, because this task is much more closely related to the target tasks. As a result, a better fit for the source task is more likely to result in better generalization across tasks.

For MLPs, there is a limited difference in performance between using the predictions or the topmost hidden layer activations as features. Sometimes the latter approach works a bit better, presumably because the feature vectors are larger (1000 values instead of 400) and sparser.

On GTZAN, we are able to achieve a classification accuracy of 0.882 ± 0.024 using the high-level features obtained from a linear regression model for the tag prediction task, which is competitive with the state of the art. If we use the low-level features directly, we achieve an accuracy of 0.851 ± 0.034. This is particularly interesting because the L2-SVM classifier is linear, and the features obtained from the linear regression model are essentially linear combinations of the low-level features.

5. CONCLUSION AND FUTURE WORK

We have proposed a method to perform supervised feature learning on the Million Song Dataset (MSD), by training models for large-scale tag prediction and user listening preference prediction. We have shown that features learned in this fashion work well for other audio classification tasks on different datasets, consistently outperforming a purely unsupervised feature learning approach.

This transfer learning approach works particularly well when the source task is tag prediction, i.e. when the source task and the target task are closely related. Acceptable results are also obtained when the source task is user listening preference prediction, although it is important to restrict the complexity of the model in this case. Otherwise, the features become too specialized for the source task,

which hampers generalization to other tasks and datasets.

In future work, we would like to investigate whether we can achieve transfer from more complex models trained on the user listening preference prediction task, and other tasks that are less closely related to the target tasks. Since a lot of training data is available for this task, using more powerful models than linear regression to learn features is desirable, especially considering the complexity of models used for supervised pre-training in the computer vision domain. This will require a different regularization strategy that takes into account generalization to other tasks and datasets, and not just to new examples within the same task, as it seems that these two do not always correlate. We will also look into whether using different dimensionality reduction techniques instead of WMF can lead to representations that enable better transfer to new tasks.

6. REFERENCES

[1] Yoshua Bengio. Learning deep architectures for AI. Technical report, Dept. IRO, Université de Montreal, 2007.

[2] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010.

[3] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, 2011.

[4] Adam Coates and Andrew Y. Ng. Learning feature representations with k-means. *Neural Networks: Tricks of the Trade, Reloaded*, 2012.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[6] Sander Dieleman and Benjamin Schrauwen. Multiscale approaches to music audio feature learning. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, 2013.

[7] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

[8] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[9] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[10] 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 *ISMIR 2013*, 2013.

[11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. Technical report, University of Toronto, 2012.

[12] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, 2008.

[13] Eric J. Humphrey, Juan P. Bello, and Yann LeCun. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, 2012.

[14] Edith Law and Luis von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *Proceedings of the 27th international conference on Human factors in computing systems*, 2009.

[15] Brian McFee, Thierry Bertin-Mahieux, Daniel P.W. Ellis, and Gert R.G. Lanckriet. The million song dataset challenge. In *Proceedings of the 21st international conference companion on World Wide Web*, 2012.

[16] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.

[17] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[18] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.

[19] Jan Schlüter and Christian Osendorfer. Music Similarity Estimation with the Mean-Covariance Restricted Boltzmann Machine. In *Proceedings of the 10th International Conference on Machine Learning and Applications (ICMLA)*, 2011.

[20] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[21] Klaus Seyerlehner, Gerhard Widmer, and Tim Pohle. Fusing block-level features for music similarity estimation. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, pages 225–232, 2010.

[22] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10:293–302, 2002.

[23] Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems 26*, 2013.

[24] Jason Weston, Samy Bengio, and Philippe Hamel. Large-scale music annotation and retrieval: Learning to rank in joint semantic spaces. *Journal of New Music Research*, 2011.

[25] J. Wülfing and M. Riedmiller. Unsupervised learning of local features for music classification. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012.

[26] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.