Spectrogram-Based Music Recommendation and Classification with Residual Convolutional Networks

Amr Tairi
Department of Computer Science
University of Maryland, College Park
atairi@umd.edu

Affan Malik
Department of Computer Science
University of Maryland, College Park
affan@umd.edu

Hrithasma Pant
Department of Computer Science
University of Maryland, College Park
hpant@umd.edu

Thomas Hyland
Department of Computer Science
University of Maryland, College Park
thyland@terpmail.umd.edu

Thomas Urdinola
Department of Computer Science
University of Maryland, College Park
email4@umd.edu

Vasu Mittal
Department of Computer Science
University of Maryland, College Park
vmittal1@terpmail.umd.edu

Abstract

Music recommendation systems often rely on metadata and user interaction patterns, potentially neglecting the intrinsic audio features that define a song. To address this limitation, we propose a two-stream Residual Convolutional Neural Network (ResCNN) architecture that processes mel-spectrogram representations of audio tracks, capturing both temporal and spectral characteristics. The temporal stream utilizes horizontal filters to extract features over time, while the spectral stream leverages vertical filters to identify frequency-based patterns. By fusing the outputs from these streams, our model achieves a robust representation of musical features. Evaluated on the GTZAN dataset, the proposed model achieves a test accuracy of 84.32%, outperforming many of the previous CNN-based approaches. Additionally, the embeddings generated by the model enable song similarity assessments, facilitating applications in music recommendation systems. This work highlights the efficacy of leveraging deep audio features for genre classification and personalized music suggestions.

1 Introduction

Music streaming platforms have revolutionized how people discover and enjoy music, offering personalized features like Spotify's Song Radio and Daylist. Existing recommendation systems fail to consistently suggest songs with comparable auditory features, such as rhythm, tempo, and frequency composition. This limitation often leads to user dissatisfaction, particularly for those seeking recommendations that align more closely with the sound of a song rather than its popularity or

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

metadata. Recognizing this gap, we propose an audio-centric approach for music recommendation by leveraging deep learning techniques. Our model processes the raw audio content of songs, focusing on both temporal and spectral features, to classify and recommend music more effectively. Through this approach we aim to offer listeners a richer and more meaningful music discovery experience, addressing the limitations of traditional recommendation systems.

2 Data

Mel-Spectrograms, representing audio signals in the time-frequency domain, are essential for capturing both temporal and spectral features critical to music classification. These 2D representations leverage time (x-axis), frequency bands (y-axis), and pixel intensity (amplitude) to encode audio patterns. Temporal patterns describe the evolution of features like rhythm or melody, while spectral patterns capture the tonal distribution of sound energy, both vital for distinguishing genres. The GTZAN dataset, consisting of 1,000 audio clips and corresponding spectrograms, is a commonly used dataset in this field of work, and due to its relatively small size, it was an ideal choice for us. To align with the baseline study [Dong, 2018], we processed the spectrograms similarly, converting audio into Mel-Spectrograms and dividing them into 3-second time segments. This ensured direct comparability with the baseline, providing a consistent reference for evaluating our model's performance.

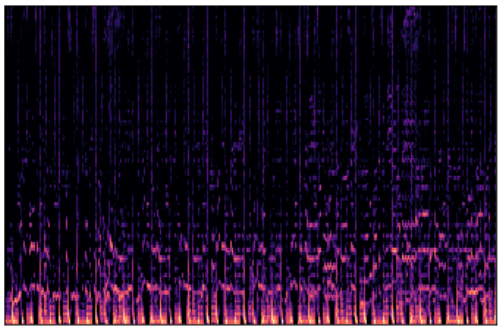


Figure 1: Spectrogram Data Example in GTZAN

3 Related Works

Music recommendation systems aim to personalize user experiences by suggesting music that aligns with individual preferences. Traditionally, these systems have relied on collaborative filtering models, used by platforms like Spotify and Netflix [Zhou et al., 2008], which predict user interests based on past interactions among users and items. However, collaborative filtering has notable limitations, including popularity bias—favoring best-sellers and widely rated items over less-known products—and the cold-start problem, where new users or items lack sufficient data for accurate recommendations [Cardorelle, 2018]. These limitations necessitate alternative approaches that can provide more nuanced and effective recommendations.

To address these issues, there is a growing shift toward content-based approaches that analyze the intrinsic audio content to measure song similarity. Unlike collaborative filtering, these methods promote diversity, require less user data, and are effective at recommending niche or new artists. A popular technique involves converting audio into Mel-Spectrograms to visualize audio features, which are then fed into Convolutional Neural Networks (CNNs) for feature extraction and classification [Cardorelle, 2018].

CNN-based models have shown significant promise in extracting audio features from spectrograms. [Sridhar, 2024] proposed an attention-guided spectrogram sequence modeling approach, emphasizing the importance of localized attention mechanisms to enhance genre classification performance. Another study [Chan et al., 2023] introduced a hybrid architecture combining CNNs and Transformers,

achieving improved classification accuracy by capturing both local and global dependencies in spectrogram data. On similar lines, [Choi et al., 2017] integrated Convolutional and Recurrent Neural Networks (CRNNs) to process both spatial and sequential information in spectrograms, addressing the limitations of purely convolutional approaches. Their works have demonstrated the effectiveness of hybrid models in handling temporal dependencies in music data, which is critical for robust genre classification.

Mingwen Dong [Dong, 2018], whose study serves as the baseline for our project, focused primarily on the temporal aspects of spectrograms, using fixed 3-second segments for genre classification. While this method achieved impressive accuracy, it often struggled with genres requiring more comprehensive spectral analysis. Our work directly builds on this limitation by introducing a dual-branch architecture to process temporal and spectral features in parallel, thus addressing the shortcomings identified in Dong's approach.

Building upon these insights, our project leverages recent advancements in CNN-based architectures, including residual connections and the parallel processing of temporal and spectral features, to classify music genres with higher accuracy. Our aim was to design a sound-focused, learning-based recommendation system that overcomes the limitations of collaborative filtering and successfully captures both local and global dependencies in spectrogram data to provide quality recommendations.

4 Method

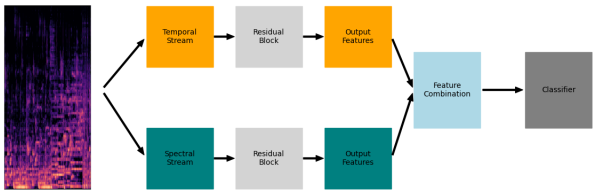


Figure 2: Model Architecture

Our work was inspired by the baseline study [Dong, 2018], which demonstrated the efficacy of using 3-second time-based segments of spectrograms for genre classification. While this approach achieved reasonable accuracy, its reliance on 3-second time segments limited its ability to fully capture temporal dynamics and nuanced features, hindering generalization across all genres. Initially, we sought to address these limitations by proposing an approach that utilized multiple segment bins of varying durations. This method aimed to capture a broader range of temporal patterns and overcome the challenges outlined in the baseline study. However, this strategy introduced practical challenges, particularly in maintaining consistent dimensionality across the segmented feature maps, complicating their integration into a unified architecture.

Having encountered challenges in successfully implementing the idea of using multiple time segments, we took the opportunity to deeply reevaluate our approach. This reflection led to a key insight: a notable limitation of the baseline paper was its inability to effectively capture the spectral features of the spectrogram. Building on this realization, we proposed a Two-Stream architecture designed to independently process both temporal and spectral features. The temporal branch utilizes horizontal filters to extract time-based patterns, while the spectral branch employs vertical filters to focus on frequency-domain characteristics. By combining the outputs of these branches, we achieved a more comprehensive and enriched feature representation, addressing the shortcomings of single-stream models and enabling a more nuanced analysis of spectrogram data.

A model's generalization is a key factor in its performance. Recognizing this, and after discussing strategies, we decided to enhance the architecture's capabilities by introducing residual connections into each branch of the Two-Stream design. This modification would not only improve the model's generalization but also would allow it to bypass certain layers during backpropagation, mitigating

vanishing gradient issues. By preserving critical low-level features and enabling deeper feature exploration, these residual connections could significantly bolster the model's learning potential.

The final model integrates these insights into a cohesive architecture. The temporal branch employs horizontal filters and a residual block to extract time-based features, while the spectral branch utilizes vertical filters and a residual block for frequency-domain analysis. The outputs from both branches are concatenated and passed through a 1×1 convolution for dimensionality reduction. The model then applies adaptive average pooling and a softmax layer to predict genre probabilities.

5 Experiments

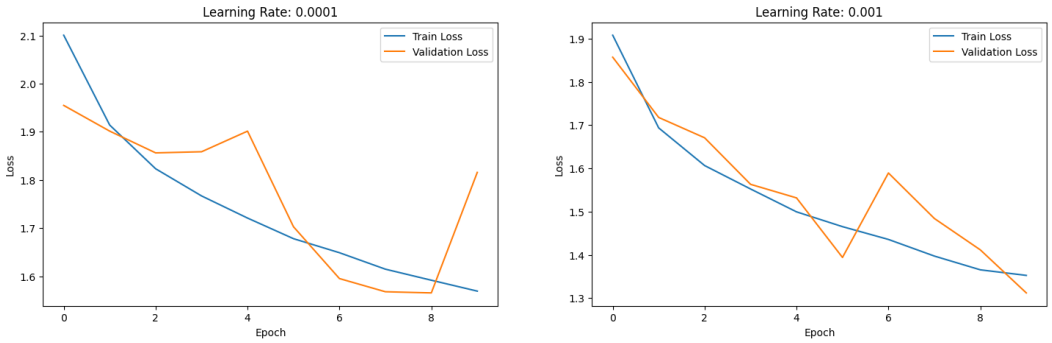
During our experiments to improve the performance of our model, we concentrated on fine-tuning three key hyperparameters: the learning rate, the number of epochs, and the patience parameter. However, based on prior experience, we decided to exclude the number of epochs from further exploration due to time constraints and resource requirements associated with running experiments across varying epoch values. Instead, our analysis centered on the impact of learning rates and the patience parameter. By systematically adjusting these variables, we analyzed their effects on the training and validation loss curves to better understand their impact on model optimization

5.1 Patience

Patience is a parameter that we used for early stopping, which would halt the training of our model when its performance on the validation set stopped improving. We experimented with patience at levels of 2, 5, 10, and 20 when fine-tuning our model, which meant the model would stop after that amount of consecutive epochs stopped improving. We did this to allow the model to continue training even when the validation loss did not improve for several consecutive epochs. This helped us to prevent premature stopping and allowed us to further experiment with using smaller learning rates. We ultimately found that patience = 10 worked the best for our model and allowed us to safeguard against over-fitting and under-training. This is further validated by the hyperparameter results stated in [Dong, 2018].

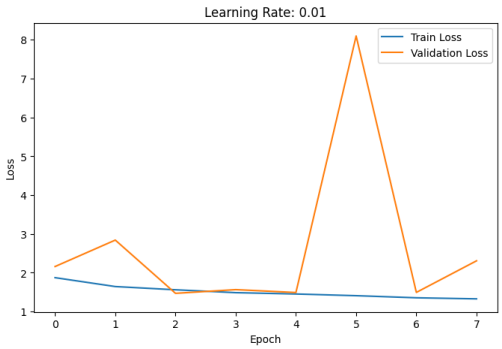
5.2 Learning Rate

The learning rate determines the size of the steps taken during the optimization process to minimize the loss function. We experimented with learning rates of 0.0001, 0.001, 0.01, and 0.1 respectively. Learning rates above 0.01 showed instability in validation loss, indicating potential over-fitting, while rates below 0.001 reduced loss too slowly without convergence. Based on this, a learning rate slightly below 0.01 (but above 0.001) is expected to balance convergence speed and stability. To balance speed and stability, we chose 0.0090 after further experiments, as it enabled rapid learning without instability or over-fitting.

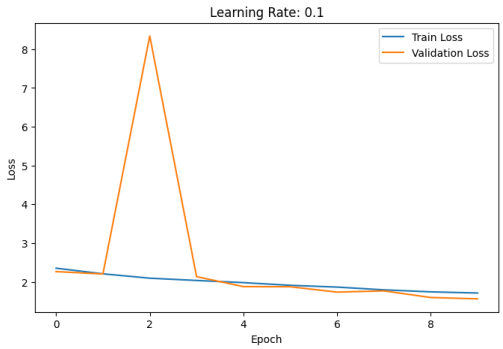


(a) Learning Rate: 0.0001 **(b)** Learning Rate: 0.001

Figure 3: Training and Validation Losses for Learning Rates: 0.0001 and 0.001.

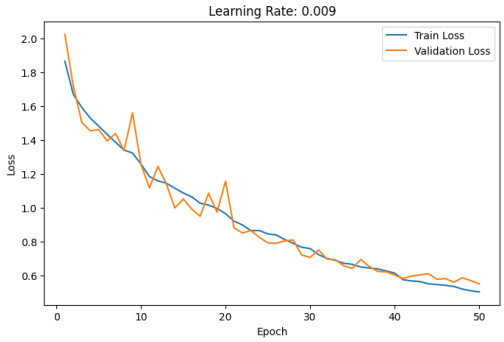


(c) Learning Rate: 0.01



(d) Learning Rate: 0.1

Figure 4: Training and Validation Losses for Learning Rates: 0.01 and 0.1.



(e) Learning Rate: 0.009

Figure 5: Training and Validation Loss for Learning Rate: 0.009.

6 Results

6.1 Comparison of Models

Our model achieved a classification accuracy of 84.32%, significantly surpassing our initial goal of 75% and demonstrating strong performance in music genre classification. Rigorous hyperparameter tuning, including optimized learning rates and early stopping, contributed to the model's ability to generalize effectively across a diverse range of musical genres.

In comparison to earlier models, such as the [Tzanetakis and Cook, 2002] framework, which achieved a 62% accuracy using handcrafted features, our model demonstrates a significant improvement due to its ability to process and learn directly from spectrogram representations. Similarly, our architecture surpasses the 75% accuracy achieved by [Cardorelle, 2018], which also utilized spectrogram inputs but lacked architectural refinements such as residual connections. Additionally, our model outperformed efficient state-of-the-art architectures like MobileNet V2 (80.37%) and transformer-based methods like the Swin Transformer (80.06%), as well as the S3T framework proposed by [Zhao et al., 2022], which achieved an accuracy of 81.10%. These comparisons highlight our model's competitive edge and its ability to extract meaningful features.

The success of our model can be attributed to its ability to balance computational efficiency with effective feature extraction through its convolutional layers. Its robustness is evident from its strong performance on challenging genres like "Classical" and "Metal," which require precise capturing of intricate patterns. These results underscore the model's capability to accurately capture the nuances of genre-specific audio characteristics while maintaining computational efficiency.

Model Name	Model Accuracy (%)
[Chan et al., 2023]	87.41
DenseNet121	86.31
ResNet18	85.51
ResNet34	84.55
Xception	84.42
Our Model	**84.32**
Inception V3	83.30
Inception V4	82.03
[Zhao et al., 2022]	81.10
MobileNet V2	80.37
EfficientNet V2 B0	80.18
Swin Transformer	80.06
MobileNet V3 small	79.88
MobileNet V3 large	78.91
EfficientNet B0	78.42
GhostNet 100	78.12
[Cardorelle, 2018]	75.00
MobileViT small	73.02
ViT	71.03
College Students Survey	70.00
[Tzanetakis and Cook, 2002]	62.00

Table 1: Music Genre Classification Model Accuracies

6.2 PCA (Principal Component Analysis) visualization - relationship between genres

We used PCA to help us visualize high-dimensional data, specifically spectrograms, by projecting them into a 2D or 3D space while keeping as much variability as possible. This approach gave us a way to see how well our model works by checking if songs cluster together correctly based on their genre in the reduced space. The clusters are distinct and don't overlap which shows the model is doing a good job of identifying genres and making solid recommendations.

We started by extracting high-dimensional embeddings (the features the model learned) for each spectrogram in our dataset and grouped them by genre. This was then averaged into genre signatures which condensed all the songs in a genre into a single representation, making it easier to analyze relationships. Once we had these signatures, we ran PCA on them to visualize their relationships in either 2D or 3D. Each genre got its own color so we could easily see how similar or different they are.

Looking at our 2D plots (Figure 4), we noticed clear distinctions between genres, which shows that the model is performing well. For example, genres like "classical" and "metal" were very distinct and far apart, meaning their embeddings are unique and well-defined. On the other hand, genres like "rock", "blues" and "country" were closer together, which makes sense since they share some overlapping features. In contrast, the 3D plots (Figure 5) added more depth to our analysis. Some genres, like "disco" and "pop," that seemed close in 2D turned out to be more distinct in 3D. This shows that the extra dimension helps capture differences that were not as obvious before. At the same time, genres like "rock", "blues" and "country" stayed close in 3D, which reflects their overlapping characteristics. While the 2D PCA captured the main differences in the data, the 3D visualization gave us a more nuanced view of how genres are related.

These visualizations are super useful for understanding how well the model is working and whether it's learning meaningful patterns. Beyond just validating our model, these embeddings have practical uses, like improving song recommendations. For example, songs with embeddings close to a genre's signature in PCA space would be recommended to someone exploring that genre. Overall, these plots are a great way to validate our model.

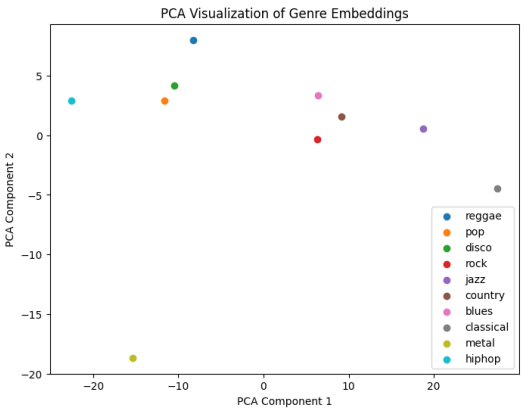


Figure 6: Genre Embeddings 2D PCA

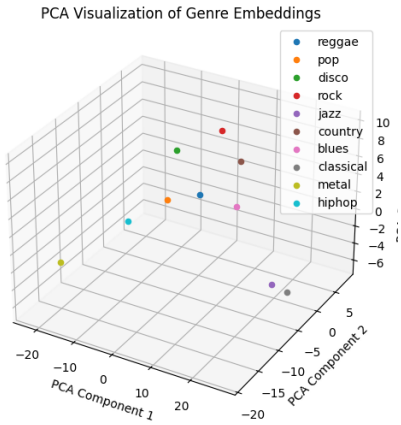


Figure 7: Genre Embeddings 3D PCA

6.3 Confusion matrix - variations in classification accuracy by genre

To gain a deeper understanding of our model beyond just accuracy or loss metrics, we utilized a confusion matrix to show how classification accuracy varies by genre. This approach made it easier to draw a comparison between the baseline model and the one we developed, as the baseline report also included a confusion matrix.

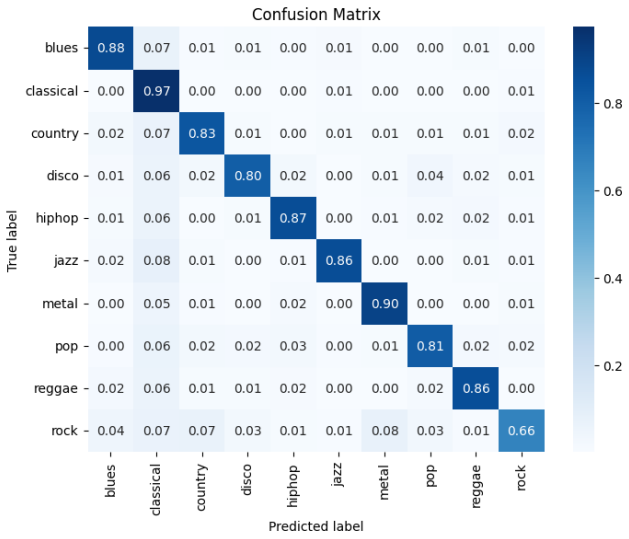


Figure 8: Confusion Matrix showing classification accuracy of our model

From the confusion matrix (Figure 8), the genres with the highest classification accuracy appear to be classical (97%), metal (90%), and blues (88%). In contrast, the rock genre had notably lower classification accuracy, with only 66% of samples correctly classified. Compared to the baseline model, our developed model outperformed it in nearly every genre, showing equal or higher accuracy in most cases, with the exception of disco.

Despite these improvements, rock remains a challenging genre to classify, possibly due to overlapping features with other genres, making it difficult to classify by nature. Thus, consulting experts in the genre could help resolve the discrepancies in classification accuracy. However, unlike the baseline model, our model appears to have resolved some of these challenges in classification accuracy. For instance, the country genre demonstrates a significantly higher classification accuracy (83%) in our model. This suggests that our model may be capturing genre-specific features more efficiently, which could be attributed to the overall architecture.

Nevertheless, the low classification accuracy of rock suggests possible areas for future improvement. One possibility could be incorporating longer audio segments or additional song features to better capture characteristics, such as beat and rhythm. With such adjustments, we could further improve the model's ability to classify rock from similar genres.

6.4 Inference & User Study

To demonstrate our model's practical application for song recommendations, we implemented a content-based music recommendation system that generates playlists of the 10 most similar songs based on audio features extracted from spectrograms. Given an input track, we passed its spectrogram through the trained model to extract its embedding. We then computed the cosine similarity between this embedding and those of all tracks in the evaluation dataset. By ranking songs based on their similarity scores, we recommended the top 10 most similar tracks. This system effectively demonstrated the model's ability to understand musical structure and style through learned feature representations.

Additionally, we conducted a user study involving 10–15 participants, who were asked to randomly select a genre and then listen to the top 10 similar tracks recommended by the system. The participants provided overwhelmingly positive feedback, noting that the beats and rhythms between the recommended tracks were highly similar, validating the system's ability to generate accurate and meaningful recommendations.

7 Conclusion & Discussion

Our model demonstrates a robust ability to classify music genres through a residual convolutional neural net architecture that captures both temporal and spectral features. We struck a balance between model complexity and performance after fine-tuning, and achieved a competitive accuracy of 84.32%. We used our model to generate a playlist of top 10 most similar songs based on an initial song and its spectrograph, resulting in recommendations based on audio-features.

Our model performed best in distinguishing genres with unique audio features, such as Classical and Metal, while broader genres like Rock and Pop were more difficult to differentiate. This highlights opportunities for future improvements, such as incorporating longer audio segments, multi-modal data such as lyrics and metadata, leveraging more extensive datasets to train on, improving attention mechanisms to the present architecture, and making use of different time based segments to extract better temporal data.

8 Future Works

While our architecture demonstrates strong performance in music genre classification, there are key areas for future improvement. Training on a more extensive and diverse dataset could enhance the model's generalizability, addressing potential biases and improving classification accuracy across a broader range of musical genres. Solving the challenge of processing longer or multiple overlapping time segments would enable the model to better capture temporal patterns, further enhancing its ability to differentiate between genres with subtle or overlapping characteristics.

Incorporating attention mechanisms into the architecture could prioritize critical temporal and spectral features, refining the model's focus on the most relevant information and boosting accuracy. Additionally, expanding on the current recommendation system, we aim to complete the development of a web application to showcase its capabilities. Conducting a larger-scale user study through this application would provide valuable feedback on its real-world usability and effectiveness, guiding future refinements.

Finally, exploring multi-modal approaches, such as integrating lyrics or metadata with audio data, could further enhance the recommendation system by providing a richer context for classification and personalization. These steps would strengthen our model's scalability and usability, advancing its practical application in content-based music recommendation systems.

References

S. Cardorelle. How i taught a neural network to understand similarities in music audio, 2018. URL https://medium.com/@silvercloud438/how-i-taught-a-neural-network-to-understand-similarities-in-music-audio-d4fca54c1aed.

J. Chan et al. A hybrid parallel computing architecture based on cnn and transformer for music genre classification. *Electronics*, 13(16):3313, 2023. URL https://www.mdpi.com/2079-9292/13/16/3313.

K. Choi et al. Convolutional recurrent neural networks for music classification. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7952585.

M. Dong. Convolutional neural network achieves human-level accuracy in music genre classification. 2018. URL https://arxiv.org/pdf/1802.09697.

A. Sridhar. Attention-guided spectrogram sequence modeling with cnns for music genre classification. 2024. URL https://arxiv.org/abs/2411.14474.

G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002. doi: 10.1109/TSA.2002.800560.

H. Zhao, C. Zhang, B. Zhu, Z. Ma, and K. Zhang. S3t: Self-supervised pre-training with swin transformer for music classification. pages 606–610, 2022. URL https://ieeexplore.ieee.org/document/9771982.

Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. pages 337–348, 2008. URL https://doi.org/10.1007/978-3-540-68880-8_32.