A Deep Learning Comparative Study for Music Genre Classification

Meredith Rush
Georgia Institute of Technology
350 Ferst Dr.
mrush30@gatech.edu

Pranav Datta
Georgia Institute of Technology
350 Ferst Dr.
datta@gatech.edu

Abstract

Large music streaming services such as Spotify face challenges in effectively recommending songs to users. Currently, these platforms rely on creating listening profiles of their users and recommending songs based on the listening histories of similar users, without leveraging the intrinsic characteristics of the music itself for determining song similarity. To offer a different method for song recommendation systems, we conducted a comparative study of seven different deep learning models tasked with predicting a song's genre based on raw audio.

For training, we used the popular GTZAN dataset which consists of 1,000 audio clips categorized by genre. To increase the size of the dataset, each clip was split into three separate audio files. Each file was then converted into a Mel-spectrogram, a graphical display of a song's frequency and loudness with respect to time. The models featured in the comparative study are the following: LSTM, Transformer encoder, 1D CNN, 2D CNN, 2D convolutional encoder, 1D convolutional encoder, and ResNet.

We tested each model with the Mel-spectrograms as the input and the target value being the respective song's genre. Our results found that the LSTM and 1D convolutional encoder models exhibited the highest accuracy at approximately 70%. Considering its high accuracy in combination with low training times and robustness against hyperparameter values, we recommend the adoption of the 1D convolutional encoder architecture for music recommendation systems.

1. Introduction

1.1. Problem and proposed solution

With the vast library of music online, people rely on recommendation systems of applications such as Apple Music and Spotify to discover new music. To offer tailored suggestions, these platforms collect extensive data to analyze the listening histories and habits of users. This recommendation system, called collaborative filtering, identifies similar users and recommends songs liked by other listeners with comparable music preferences.

This reliance on the behavior of users presents a multitude of problems, namely, the ”cold start” and ”echo chamber” problems. The former arises when there is insufficient historical data about a user or a song. Because their respective listening profiles have not been curated yet, new users do not receive accurate recommendations and, similarly, new songs are not recommended. The ”cold start” problem for users is particularly emphasized for smaller music recommendation applications, as the user base is dominated by Apple Music and Spotify. The ”cold start” problem for songs results in a bias toward recommending popular songs, contributing to the ”echo chamber” effect where only the same songs are being circulated in recommendations. Lesser-known music consequently has a more difficult time gaining exposure.

Because collaborative filtering relies solely on user behavior, two songs could be identified as similar and consequently recommended even if they aren't audibly similar or fall within the same genre. Though Spotify has begun to use Convolutional Neural Networks to group similar songs based on their quantifiable characteristics such as valence, tempo, loudness, etc., this process still doesn't result in better song recommendations [1]. In fact, only about 44% of songs recommended by Spotify in their personalized Discover Weekly playlist are added by users [2].

These issues stem from the recommendation systems having a poor representation of songs that don't actually use the music itself, only the surrounding information about a song and its listeners. Therefore, we hypothesized that these issues would be resolved by using deep learning models on the audio of the songs for music recommendation. Using the accuracy of genre classification as the metric for determining similarity between songs, we evaluated and compared the performance of 7 types of neural networks that are trained on the sound itself to find the architecture that is most suitable for improving recommendation software.

1.2. Implications of success

We suggest that finding a method that focuses on the auditory features of songs could yield a better internal representation of the music. This, in turn, could enhance the identification of similar songs and lead to better recommendations. This approach would free music applications from needing a large network of users with rich listening histories, alleviating the ”cold start” and ”echo chamber” problems. With these models, the user-base dominance of Apple Music and Spotify would no longer hinder other prospective music applications, as having a large number of users would not be a necessity for producing satisfactory recommendations.

Since these models have no popularity bias, lesser-known songs would get more exposure if they were objectively more similar to songs a user enjoys. The objectivity inherent in these models increases the merit of two songs being identified as similar, as the similarity is a result of the intrinsic qualities of the songs themselves rather than the backgrounds of their listeners. Determining the optimal model for music genre classification through our experiments would provide a guide for the architectural direction this type of recommendation system should take.

2. Related Works

Music genre recognition poses unique challenges due to their subjective nature. George Tzanetakis and Perry Cook justify the use of genres, explaining that although genres may be classified differently by various people, there exists a discernible pattern in pitch and structure within genres of music [3]. With this understanding, numerous experiments have utilized deep learning models for music genre classification, primarily with audio clips converted to a Mel-spectrogram format. The following subsections will delve into various models developed for music genre classification.

2.1. Long Short-Term Memory

Long Short-Term Memory (LSTM) models are a type of Recurrent Neural Network (RNN) that can learn long-term dependencies. Given that audio clips span over timesteps, a deep learning model that can leverage temporal information throughout a song could enhance accuracy in classifying genres. In a comparative study by Jia Dai et al., various LSTM models for music genre classification were examined, with the most accurate achieving 89.71% accuracy [4].

2.2. Transformer Encoder

An encoder-only transformer, a variant of transformers that only contains the encoder, is utilized for comprehending full sequences. Encoder transformers have been applied to music genre classification tasks, particularly by analysis of song lyrics using natural language processing. An example is MusicBERT, a variation of BERT [5] that is trained with symbolic music data, which achieved an accuracy of 73% for genre recognition [6].

2.3. Convolutional Neural Networks

Convolution Neural Networks (CNN) are widely recognized for their efficacy in image classification and have gained popularity for music genre recognition. One such example is the CNN created by Yu-Huei Cheng et al. for an experiment utilizing the GTZAN dataset for music genre recognition [7]. The 2D CNN comprised five convolutional, max-pooling, and dropout layers with ReLU serving as the activation function. The model demonstrated strong performance with an accuracy of 83.30%.

In addition to 2D CNNs, research has explored the efficacy of 1D CNNs for signal processing. Signal processing involves tasks akin to sampling audio and detecting the audio's pitch over time. 1D CNNs have demonstrated state-of-the-art results for signal processing with minimal complexity [8]. Given their proficiency in signal processing, 1D CNNs could be applied to music genre classification tasks when analyzing song's frequencies over time.

2.4. Convolutional Encoder

A convolutional encoder-decoder (CED) is a transformer model where either the encoder, decoder, or both contain convolutional layers. A well-known CED model is SegNet, typically employed for semantic pixel-wise segmentation. SegNet's architecture includes a 13-layer convolutional encoder network, a corresponding decoder, and a classification layer. SegNet performs exceptionally well for image classification tasks, achieving high accuracy while minimizing the model complexity and runtime [9].

2.5. Residual Neural Network

While CNN models have been dominant for music genre recognition, other deep learning methods are gaining popularity, such as Residual Neural Networks (ResNets). A ResNet is a deep convolutional neural network capable of supporting thousands of layers, thanks to its inclusion of residual blocks. These blocks enable the model to skip layers that may hinder the model's performance. Additionally, residual blocks address the “vanishing gradient” problem encountered when training deep CNNs [10]. Recent research explores whether ResNet can outperform conventional CNNs. Junfei Zhang conducted an experiment combining a ResNet18 with a Bi-Directional Gated Recurrent Network (Bi-GRU) for music genre classification, achieving an accuracy of 90.00% [11].

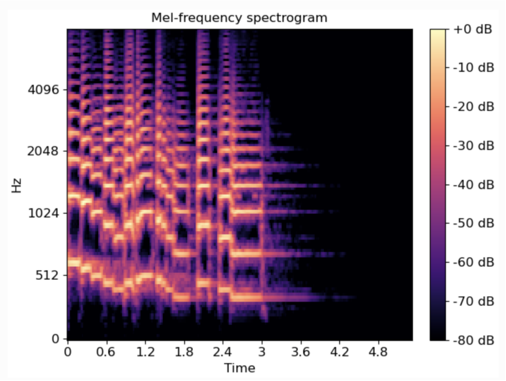


Figure 1. Mel-spectrogram as an image

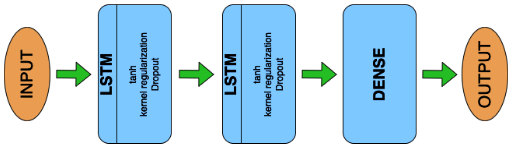


Figure 2. LSTM architecture

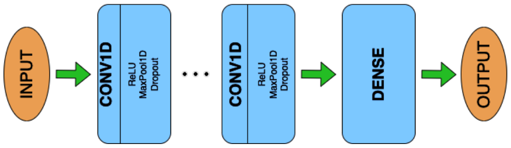


Figure 3. 1-dimensional CNN architecture

3. Method and Approach

3.1. Inputs

Our dataset contained 1000 30-second song clips spanning 100 genres. To increase the dataset size, each song was split into three 10-second clips. We then converted the raw audio files to Mel spectrograms using the Librosa Python library and one-hot encoded the genre labels to serve as the inputs for our models. We deliberately avoided further data augmentation out of concerns that altering the spectrograms may transform the song beyond its genre classification.

Mel spectrograms are 2-dimensional matrix representations of audio, with timesteps along one axis and frequency along the other. Each entry's magnitude signifies the strength of a frequency at a particular timestep [12]. A visual representation of this matrix can be seen in Figure 1. Although Mel spectrogram conversions are a lossy compression of audio, the Mel spectrum filter is specifically designed to be selective toward frequencies processed by the human ear, rendering the data lost largely superfluous [7].

While there exist neural networks capable of processing raw audio files directly, these networks must be complex to achieve satisfactory accuracy, leading to slower runtimes. One such network, WaveNet, is 60 layers deep and takes minutes to process a second of audio [12]. Our computational resources were insufficient to run such a model, hence we preprocessed our audio inputs to have lower dimensionality suitable for our models.

3.2. LSTM

As spectrograms involve temporal information, we decided to begin with LSTM networks due to their effectiveness in processing sequential data. The general architecture was inspired by sequence-to-sequence encoders, featuring 2 LSTM layers with Tanh activations. To mitigate overfitting, we added kernel regularization and dropout after each LSTM layer. Figure 2 illustrates our LSTM architecture.

While RNNs could have been a viable alternative due to their proficiency in processing sequences, the numerous timesteps of song data (about 430 timesteps per clip) would have rendered the model prone to the vanishing or exploding gradient problems. LSTMs, on the other hand, are less susceptible to these issues, enabling them to learn long-term dependencies over many timesteps [13]. Furthermore, studies have shown a statistically significant relationship between LSTMs and brain activity, suggesting their potential to develop internal representations of music akin to human cognition [14].

3.3. Transformer encoder

However, LSTMs exhibit diminished performance with longer sequences, as contextual information decreases exponentially with distance. Additionally, they are also generally slow because they process sequentially and not in parallel [15]. As spectrograms can be considered as longer sequences, Transformer encoders may be more suitable for the music genre classification task as they circumvent the long-range dependency and parallelization problems with multi-head attention [16]. In our application of the Transformer encoder, we included dropout to mitigate overfitting.

3.4. 1-dimensional CNN

Given their efficacy in signal processing, we deemed 1-dimensional CNNs to be suitable for processing audio signals [8]. A kernel that slides along one axis aligns with the nature of time-series data such as music [17]. Drawing inspiration from the architecture introduced by Yu-Huei Cheng et al. using 2-dimensional CNNs, our implementation used 1-dimensional CNNs featuring multiple blocks comprising Conv1D layers with ReLU activations followed by max pooling and dropout layers [7]. The inclusion of max pooling layers serves the dual purpose of forwarding only the most important features to subsequent layers while also reducing dimensionality. Figure 3 illustrates our 1-dimensional CNN architecture.

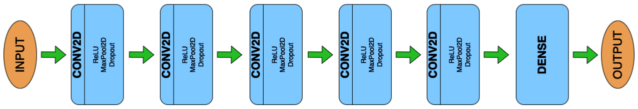


Figure 4. 2-dimensional CNN architecture

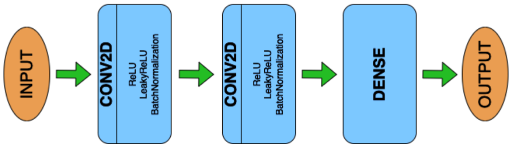


Figure 5. 2-dimensional convolutional encoder architecture

3.5. 2-dimensional CNN

Since spectrograms can be represented as images, we reasoned that 2-dimensional CNNs may achieve high accuracy due to their strength in image classification. Using this method, Yu-Huei Cheng et al. were able to reach 83% accuracy in the music genre classification task [7]. We used a variation of their architecture as the basis for our 2-dimensional CNN. Similar to the structure of the 1-dimensional model, our design comprised 5 blocks of Conv2D layers with ReLU activations followed by max pooling and dropout layers. Figure 4 illustrates our 2-dimensional CNN architecture.

3.6. 2-dimensional convolutional encoder

2-dimensional convolutional encoders, such as the one exemplified in the SegNet architecture, have demonstrated the ability to discern semantic relationships among various features to delineate between objects in an image [9]. We assessed whether such a model could identify semantic relationships within spectrogram images to achieve optimal performance in the music genre classification task. The architecture for our 2-dimensional convolutional encoder consisted of 2 Conv2D layers with ReLU activations, each followed by leaky ReLU and batch normalization layers. Figure 5 illustrates our 2-dimensional convolutional encoder architecture.

3.7. 1-dimensional convolutional encoder

With SetNet's ability to represent complex data at a lower dimensionality, we explored whether a 1-dimensional convolutional encoder, inspired by the SegNet architecture, could similarly identify semantic relationships among spectrograms of songs within the same genre. The general architectures for our 1-dimensional convolutional encoders comprised multiple blocks of Conv1D layers with ReLU activations followed by leaky ReLU and batch normalization layers. Figure 6 illustrates our 1-dimensional convolutional encoder architecture.

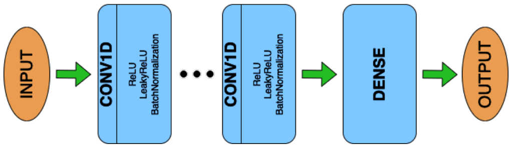


Figure 6. 1-dimensional convolutional encoder architecture

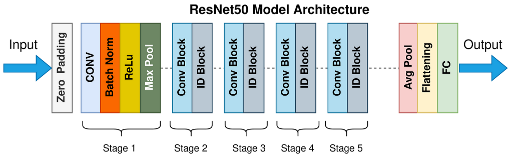


Figure 7. ResNet50 architecture

3.8. ResNet

The previously discussed models are relatively shallow, especially the 1-dimensional CNNs which typically require many Conv1D layers to enhance performance. Therefore, we aimed to evaluate the impact of a deeper model on the music genre classification task. ResNets are neural networks known for incorporating a large number of layers while still maintaining optimal performance. ResNets contain a convolutional layer followed by multiple residual blocks, each containing 2 convolutional layers [10]. We evaluated the ResNet50, ResNet101, and ResNet50V2 models. Figure 7 illustrates the architecture of ResNet50.

3.9. Outputs

Each model includes a fully-connected layer with softmax activation, generating probabilities representing the model's predictions for each genre. We used categorical cross entropy, or softmax loss, as our accuracy metric, given the nature of the multiclass classification task over genres [18].

4. Data

Our models are tested on the GTZAN dataset, created by Georgia Tzanetakis and Perry Cook [3]. The dataset comprises 1,000 audio recordings distributed evenly across ten genres: Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Rock, and Reggae. Recorded between 2000-2001, the audio samples originate from various sources such as CDs, radio, and microphone recordings. Each audio recording is 30 seconds long and is stored in a .wav file. The dataset categorizes all audio files into folders labeled by their respective genres. GTZAN is a widely adopted dataset for training and evaluating music genre recognition models.

A study by Bob Sturm outlines several mislabels and redundancies in the GTZAN dataset [19]. Sturm identified 93 mislabels but acknowledged challenges in categorizing songs that span multiple genres. For example, Billy

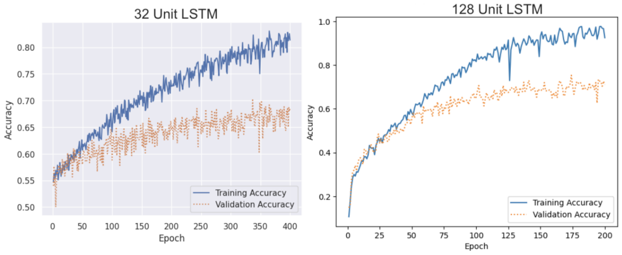


Figure 8. LSTM accuracy graphs showing overfitting

Ocean's "Can You Feel It" is labeled as disco in the dataset but could also be classified as rock or pop. Additionally, there are 13 instances of version repetitions, where different audio recordings correspond to various portions of the same songs [19]. Despite the limitations of the dataset, we anticipated minimal impact on the accuracy of our models.

In the future, alternative datasets such as Google's AudioSet [7] and Million Song Dataset [20] could complement the GTZAN dataset. When training our models, our preference was to use raw audio clips for Mel-spectrogram conversion. Most publicly available audio recording datasets provide either the original audio or genre tags, but not both. The GTZAN dataset, widely employed in genre classification models, met both criteria. Therefore, we considered it the most suitable dataset for our purposes.

5. Experiments and Results

5.1. LSTM

Determining the optimal number of hidden units for LSTMs involves extensive experimentation. We initially started with 10 hidden units, corresponding to the number of genres for classification, yielding only 35% accuracy. We achieved better results by increasing the number of hidden units to 128. While this model reached 72% accuracy in 200 epochs, training took 4 hours. In contrast, using 32 hidden units resulted in a comparable accuracy of 67%, with training completed in just 1 hour across 400 epochs. Regardless of the number of hidden units, the LSTMs were prone to overfitting, as seen in Figure 8.

5.2. Transformer encoder

Contrary to expectations, Transformer encoders performed worse than LSTMs. Despite configuring the Transformer encoder with the same number of hidden units as our most successful LSTM model and 10 heads to reflect the number of genres, it only achieved 50% accuracy. Even with a 0.5 dropout rate, the encoder began to overfit after approximately 50 epochs, as depicted in Figure 9. Notably, the Transformer encoder exhibited faster training than the LSTM with equivalent hidden units, completing 200 epochs in 2 hours as opposed to 4.

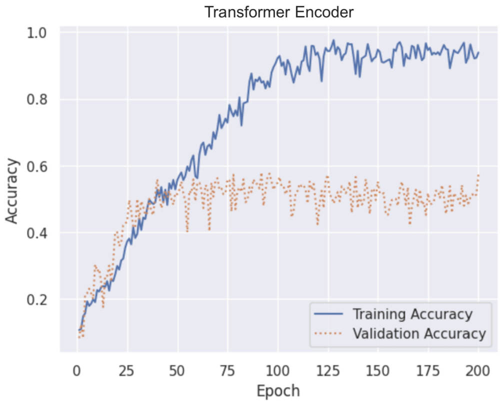


Figure 9. Transformer encoder accuracy graph showing overfitting

5.3. 1-dimensional CNN

We initiated the model with 3 Conv1D, max pooling, and dropout blocks. We configured the hyperparameters filters, kernel size, and dropout rate based on values from the Yu-Huei Cheng et al. paper: 128, 64, and 32 filters for the Conv1D layers, kernel size of 3, and dropout rate of 0.3 [7]. Despite completing 400 epochs in 30 minutes, the model only achieved approximately 25% validation accuracy, displaying signs of overfitting to the training data.

Doubling the number of filters did improve validation accuracy by 10%, but overfitting persisted. Attempts to mitigate overfitting with regularization proved ineffective, causing the model to only predict one genre.

We removed the last convolutional block to determine whether the overfitting was a result of the number of layers. Surprisingly, this adjustment led to a validation accuracy of 60%, challenging the convention that 1-dimensional CNNs require many Conv1D layers for optimal performance. However, the model was still prone to overfitting with a training accuracy of 90%. The reduced number of layers led to quicker training, completing 400 epochs in 25 minutes.

Doubling the number of filters again failed to yield further improvement and still exhibited overfitting. A comparison of the accuracies and confusion matrices during training is provided in Figures 10 and 11. Interestingly, all 1-dimensional CNNs displayed a bias toward classifying songs as disco.

5.4. 2-dimensional CNN

We initially adopted the architecture outlined in the Yu-Huei Cheng et al. paper, featuring 5 blocks of Conv2D layers followed by max pooling and dropout layers [7]. The Conv2D layers had filter counts of 128, 64, 32, 16, and 8 with a kernel size of 3 and a dropout rate of 0.5. Running for 400 epochs in 54 minutes, this model achieved 65% accuracy without overfitting.

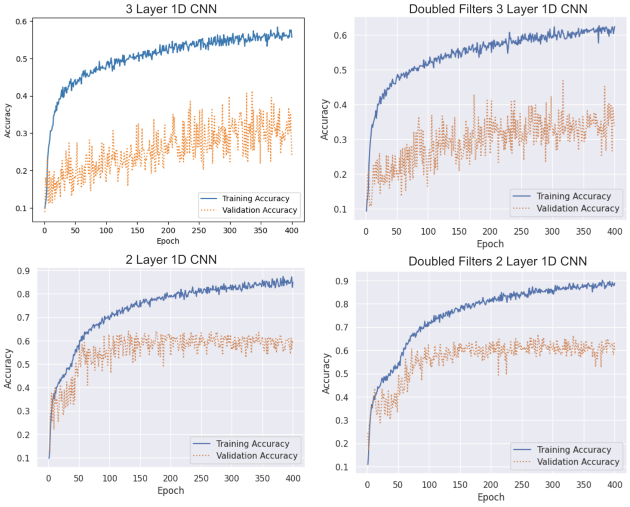


Figure 10. 1-dimensional CNN accuracy graphs

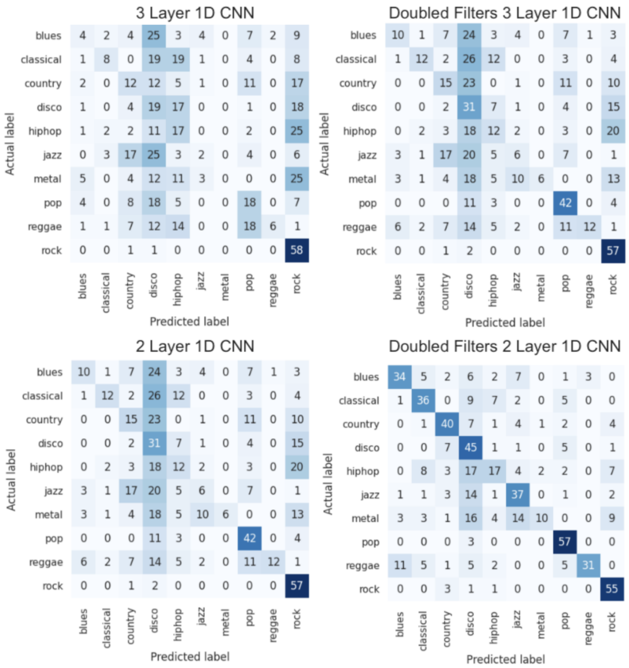


Figure 11. 1-dimensional CNN confusion matrices

Unlike the 1-dimensional CNNs, doubling the number of filters did not improve the model's accuracy. In fact, the modified model took an additional 40 minutes to complete the same number of epochs, with no substantial improvement in accuracy. Although it reached 65% validation accuracy faster, the model also began to overfit, hindering further accuracy improvement with additional epochs (see Figure 12). In contrast to the first model, which showed potential for improvement with more epochs, training for the second model appeared to plateau.

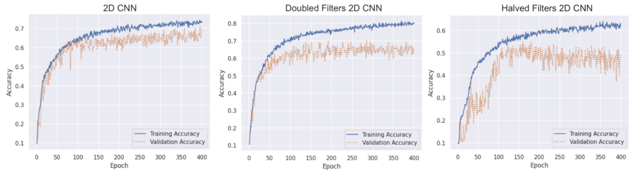


Figure 12. 2-dimensional CNN accuracy graphs

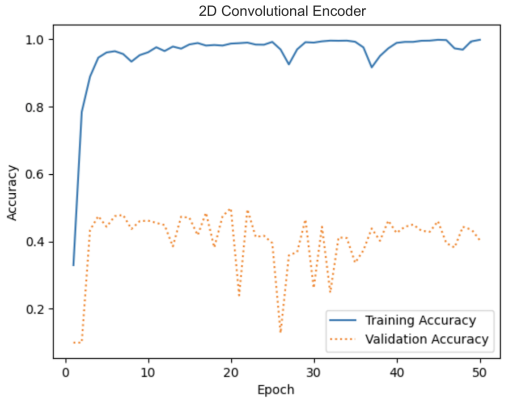


Figure 13. 2-dimensional convolutional encoder accuracy graph showing overfitting

Reducing the number of filters by half also failed to boost accuracy. This model was also prone to overfitting which caused the validation accuracy to degrade from 55% to 45% over 250 epochs (see Figure 12). However, it completed 400 epochs much faster at 38 minutes.

5.5. 2-dimensional convolutional encoder

The 2-dimensional convolutional encoder struggled with the music genre classification task. With drastic overfitting within the first 10 epochs (see Figure 13), the model was only able to reach a maximum of 50% validation accuracy while achieving 99% training accuracy. This discrepancy suggests that the SegNet architecture may be specifically designed for image segmentation and its application to music genre classification is unfitting.

5.6. 1-dimensional convolutional encoder

In contrast, the 1-dimensional convolutional encoder, inspired by SegNet, performed well at the music genre classification task, as depicted in the confusion matrix shown in Figure 14. Employing 2 blocks of Conv1D layers followed by leaky ReLU and batch normalization yielded a consistent 70% accuracy regardless of the number of filters in the Conv1D layers. In our initial test, where we used 512 filters in the first Conv1D layer and 256 in the second, the model achieved this accuracy after running for 100 epochs in 24 minutes. Interestingly, doubling or halving the number of filters maintained the same accuracy within the same time-

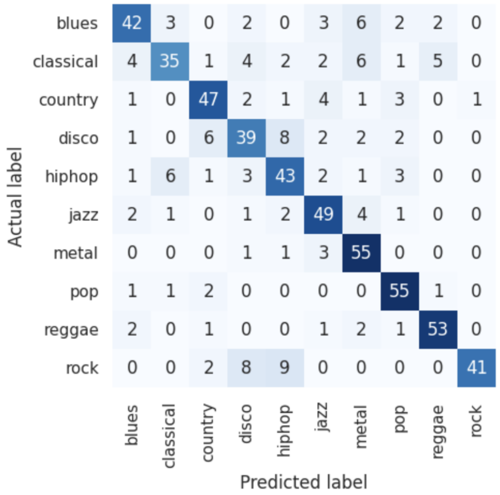


Figure 14. 1-dimensional convolutional encoder confusion matrix

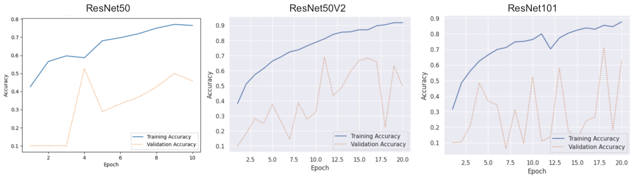


Figure 15. ResNet accuracy graphs showing overfitting

frame and epoch count.

In an attempt to further improve accuracy, we added additional blocks to the model. However, this led to severe overfitting with training accuracies reaching 100%.

5.7. ResNet

Due to their depth, training the ResNet models required a considerable amount of time for each epoch. The ResNet50 model, taking 20 minutes for 10 epochs, reached a maximum validation accuracy of only 50%. There was no evidence of improvement, as degradation caused by overfitting became evident. The ResNet101 model faced similar challenges, exhibiting overfitting and a failure to converge on validation accuracy after 20 epochs, which took 1 hour. While the ResNet50V2 demonstrated the best performance, it also dealt with degradation due to overfitting. The overfitting faced by these three models are illustrated in Figure 15. Interestingly, both ResNet50 and ResNet50V2 displayed a bias towards classifying songs as jazz (see Figure 16).

6. Conclusion

6.1. Summary

Overall, none of our models achieved exceptional accuracy. Ideally, we aimed for models surpassing 80% or 90% accuracy, providing strong evidence that neural networks trained on audio could outperform existing methods in identifying music similarity. Our experimentation fell short, with the highest validation accuracy reaching only 72%. However, this does not conclusively imply that these neural networks do not develop insights about music similarity.

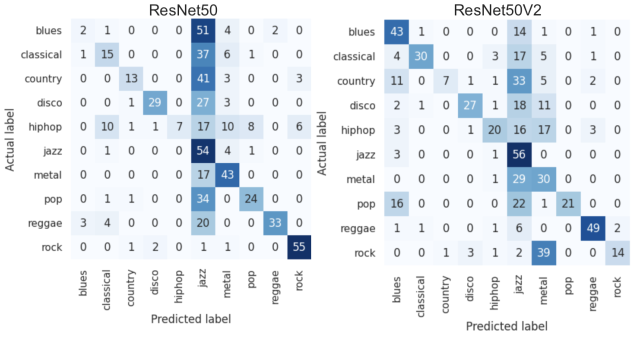


Figure 16. ResNet50 and ResNetV2 confusion matrices

Our metric for music similarity was based on genre classification accuracy. Music genres, being subjective human classifications, pose challenges in defining precise similarities between songs. While humans excel at determining song similarity, articulating these similarities can be difficult. Songs can be similar to multiple others across various genres, rendering genres as an oversimplification. Therefore, classifying a song into a single genre can be a difficult task for humans, so neural networks may face similar difficulties, explaining the observed unfavorable performance [21].

In our experiments, architectures employing 2 1-dimensional convolutional layers performed most optimally, achieving 60-70% accuracy after just 25 minutes of training, regardless of hyperparameter values. Models incorporating more convolutional layers exhibited lower accuracies and more overfitting. This was exemplified particularly with the ResNet architectures, contrary to expected behavior.

While LSTMs demonstrated comparable accuracy, they require more parameters to capture long-term dependencies that result in extended training times due to the increased complexity. In general, LSTMs are slow because they process data sequentially. These increased training times do not translate into substantial accuracy improvements. Consequently, despite their comparable performance to 1-dimensional convolutional encoders, LSTMs may not be justified for inclusion in recommendation system architectures. Transformer encoders, posited as improvements over LSTMs, did not exhibit superior performance in terms of speed or accuracy in our experiments.

Depending on hyperparameter values, 2-dimensional convolutional encoders achieved accuracy and training speeds comparable to our best models. However, their sensitivity to hyperparameters may render them less dependable for music recommendation systems.

6.2. Suggestions

From our findings, it was evident that the most effective architecture for music genre classification is the 1-dimensional convolutional encoder. Its combination of high accuracy, low training times, and resilience to varied hyperparameters may suggest that these models develop optimal internal representations of music, making them well-suited for recommender systems. However, despite being the top performer in our experiments, its suboptimal accuracy implies room for future improvements.

For future research, we recommend developing tools capable of identifying different instruments or vocals and their utilization in songs. Manipulating spectrograms to incorporate this additional information can provide the neural network models with more to extrapolate from. This modification can be tested on our models to reveal if it leads to higher accuracy, indicating that the additional information leads to improved inferences about song similarity.

Alternatively, given the subjective nature of genre classification, we suggest exploring different metrics for determining music similarity. Spotify gathers data on a song's energy, tempo, danceability, acousticsness, etc [1]. A combination of these metrics can be used to create more nuanced labels that are more representative of song similarity.

Finally, another avenue worth exploring is clustering, which can serve as an unbiased method for gauging music similarity. A recommender system using a model unrestricted by labels could be a true framework for assessing song similarity, as the model is allowed to draw on its own inferences rather than being constrained by subjectively assigned labels.

References

[1] U. C. Goodness (Goodnessuc), "Decoding how spotify recommends music to users," MUO, 07 2023. [Online]. Available: https://www.makeuseof.com/decoding-how-spotify-recommends-music-to-users/ 1, 8

[2] "Discovery mode – spotify for artists," artists.spotify.com. [Online]. Available: https://artists.spotify.com/en/discovery-mode 1

[3] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002. 2, 4

[4] J. Dai, S. Liang, W. Xue, C. Ni, and W. Liu, "Long short-term memory recurrent neural network based segment features for music genre classification," in *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2016, pp. 1–5. 2

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. 2

[6] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu, "Musicbert: Symbolic music understanding with large-scale pre-training," 2021. 2

[7] Y.-H. Cheng, P.-C. Chang, and C.-N. Kuo, "Convolutional neural networks approach for music genre classification," IEEE Xplore, p. 399–403, 11 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9394067 2, 3, 4, 5

[8] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, and M. Gabbouj, "1-d convolutional neural networks for signal processing applications," IEEE Xplore, p. 8360–8364, 05 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8682194 2, 3

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, 12 2017. 2, 4

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv (Cornell University)*, 12 2015. 2, 4

[11] J. Zhang, "Music genre classification with resnet and bi-gru using visual spectrograms," 2023. 2

[12] L. Wyse, "Audio spectrogram representations for processing with convolutional neural networks," *arXiv (Cornell University)*, 06 2017. 3

[13] C. Olah, "Understanding lstm networks," Github.io, 08 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ 3

[14] M. Hashemzadeh, G. Kaufeld, M. White, A. E. Martin, and A. Fyshe, "From language to language-ish: How brain-like is an lstm's representation of nonsensical language stimuli?" *arXiv (Cornell University)*, 01 2020. 3

[15] G. Giacaglia, "Transformers," Medium, 05 2023. [Online]. Available: https://towardsdatascience.com/transformers-141e32e69591 3

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," arXiv.org, 12 2017. [Online]. Available: https://arxiv.org/abs/1706.03762 3

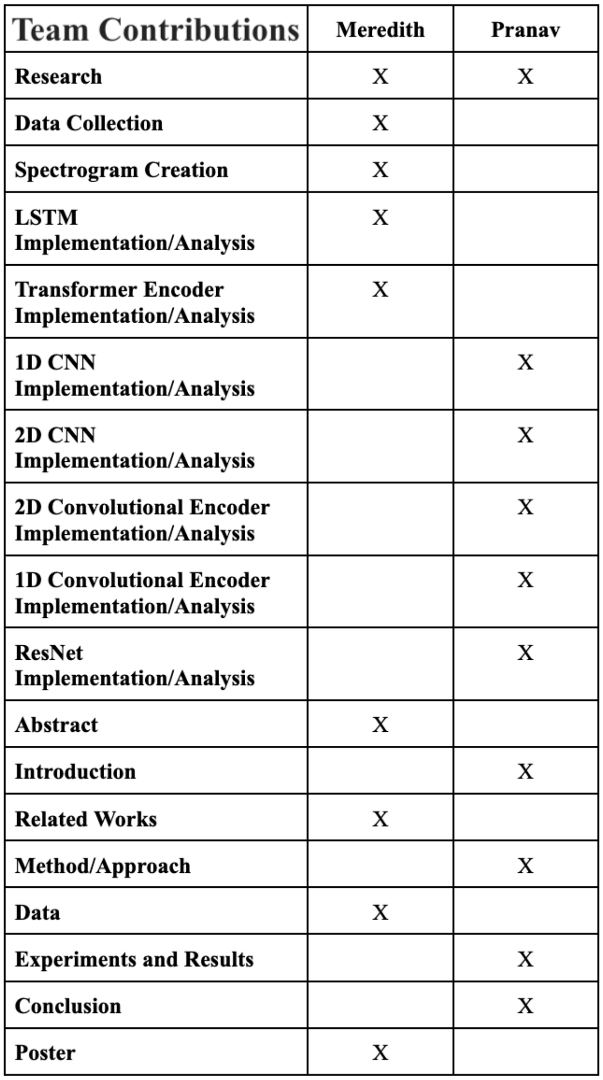
[17] A. P. Wibawa, A. B. P. Utama, H. Elmunsyah, U. Pujianto, F. A. Dwiyanto, and L. Hernandez, "Time-series analysis with smoothed convolutional neural network," *Journal of Big Data*, vol. 9, 04 2022. 3

[18] D. Shah, "Cross entropy loss: Intro, applications, code," www.v7labs.com, 01 2023. [Online]. Available: https://www.v7labs.com/blog/cross-entropy-loss-guide 4

[19] B. L. Sturm, "The state of the art ten years after a state of the art: Future research in music information retrieval," *Journal of New Music Research*, vol. 43, no. 2, p. 147–172, Apr. 2014. [Online]. Available: http://dx.doi.org/10.1080/09298215.2014.894533 4, 5

[20] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011. 5

[21] S. Allamy and A. L. Koerich, “1d cnn architectures for music genre classification,” *arXiv (Cornell University)*, 05 2021. 7



Team Contributions	**Meredith**	**Pranav**
Research	X	X
Data Collection	X	
Spectrogram Creation	X	
LSTM Implementation/Analysis	X	
Transformer Encoder Implementation/Analysis	X	
1D CNN Implementation/Analysis		X
2D CNN Implementation/Analysis		X
2D Convolutional Encoder Implementation/Analysis		X
1D Convolutional Encoder Implementation/Analysis		X
ResNet Implementation/Analysis		X
Abstract	X	
Introduction		X
Related Works	X	
Method/Approach		X
Data	X	
Experiments and Results		X
Conclusion		X
Poster	X	