Deep Learning-Based Music Genre Classification

Runzhe Huang
Khoury College of Computer Science
Northeastern University
Boston, MA, US
huang.run@northeastern.edu

Kaito Minami
Khoury College of Computer Science
Northeastern University
Boston, MA, US
minami.k@northeastern.edu

1 Introduction

1.1 Project Objectives and Problem Statement

The primary objective of this project is to develop a deep learning-based system that accurately classifies music tracks into distinct genres (e.g., rock, pop, classical, jazz, hip-hop). The model will take an audio file as input and predict the most probable genre based on the extracted features from the audio.

The problem our team aims to tackle is the accurate classification of music tracks into various genres using deep learning techniques. Music genre classification is a challenging task due to the complex nature of audio signals, which contain overlapping and subtle features. Successfully addressing this problem will enable the development of a robust system that can automatically identify the genre of a given audio file.

1.2 Significance of the Problem

Music genre classification is essential for improving user experience on streaming platforms by enabling personalized recommendations and efficient content organization. As music libraries grow, automatic classification helps users discover relevant content more easily.

From a technical standpoint, genre classification is challenging due to the complex and overlapping nature of audio features. Deep learning offers an exciting opportunity to enhance accuracy by mimicking human auditory perception, meeting the demand for more effective music categorization.

2 Background

2.1 Previous Work and Research

A study, published in the Data Science Blogathon on Analytics Vidhya by Sawan Rai, used the GTZAN dataset (containing 1000 tracks across 10 genres) to classify music genres with Convolutional Neural Networks (CNNs). Spectrograms and wavelet transforms were used to preprocess audio data, which were then input into CNNs. The study showed promising results in genre classification, although challenges were noted, particularly in distinguishing genres like rock and reggae.

Another article by Prem Tibadiya, published on Medium, used Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks to classify music genres. This study highlighted the capability of RNNs and LSTMs in capturing temporal dependencies in music, achieving notable accuracy improvements.

2.2 Challenges and Successes in Related Work

CNN-based models have seen success in classifying genres like classical music but struggle with other genres. Transfer learning has been explored to improve performance, though the gains are often minor. Class imbalance and the complexity of music features remain significant challenges, and multi-modal approaches, combining spectrogram and wavelet data, have not consistently yielded better results.

However, RNNs and LSTMs still face challenges in distinguishing between genres with similar rhythmic and harmonic features, such as rock and pop. Moreover, the computational complexity of training these models remains a barrier to achieving real-time classification in streaming applications.

3 Methods

3.1 Data

3.1.1 GTZAN Dataset

Diversity and Volume: The GTZAN dataset offers a balanced dataset for initial experiments, providing a wide range of genres and a substantial number of tracks. It includes 1,000 audio samples, evenly distributed across 10 genres (such as rock, pop, classical, jazz, and blues), with each track lasting 30 seconds. This uniformity ensures consistency in model training and evaluation, making it a great resource for genre classification tasks.

Structure and Application: One file (30 seconds dataset) contains for each song (30 seconds long) a mean and variance computed over multiple features that can be extracted from an audio file. Another file (3 seconds dataset) has the same structure, but the songs were split into 3-second audio files (increasing the amount of data by 10 times for input into our classification models). In addition, there is another dataset that contains the original 30-second audio files along with their corresponding Mel spectrograms.

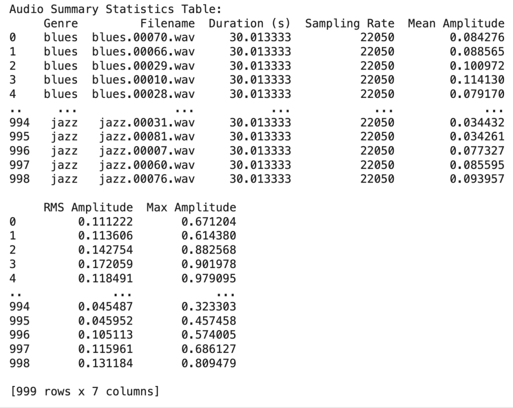
The original 30-second audio files are primarily used to extract features, as the current dataset contains a limited number of features with relatively low dimensionality. Combining the newly extracted features from the audio files with the existing features from the dataset could introduce inconsistencies due to differences in feature representations, potentially leading to overfitting or increased model complexity.

Given that the 3-second files are derived by splitting the original 30-second files, so the primary objective is to classify the 30-second one. Additionally, the 3-second dataset lacks audio files, which prevents us from extracting sufficient features necessary for deep

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

learning approaches. This makes the original 30-second audio files the most reliable source for our feature extraction and classification task.



```
Audio Summary Statistics Table:
     Genre         Filename  Duration (s)  Sampling Rate  Mean Amplitude
0    blues  blues.00070.wav     30.013333          22050        0.084276
1    blues  blues.00066.wav     30.013333          22050        0.088565
2    blues  blues.00029.wav     30.013333          22050        0.100972
3    blues  blues.00010.wav     30.013333          22050        0.114130
4    blues  blues.00028.wav     30.013333          22050        0.079170
..     ...              ...           ...            ...             ...
994   jazz   jazz.00031.wav     30.013333          22050        0.034432
995   jazz   jazz.00081.wav     30.013333          22050        0.034261
996   jazz   jazz.00007.wav     30.013333          22050        0.077327
997   jazz   jazz.00060.wav     30.013333          22050        0.085595
998   jazz   jazz.00076.wav     30.013333          22050        0.093957

     RMS Amplitude  Max Amplitude
0         0.111222       0.671204
1         0.113606       0.614380
2         0.142754       0.882568
3         0.172059       0.901978
4         0.118491       0.979095
..             ...            ...
994       0.045487       0.323303
995       0.045952       0.457458
996       0.105113       0.574005
997       0.115961       0.686127
998       0.131184       0.809479

[999 rows x 7 columns]
```

Figure 1: Audio Summary Statistics Table

Also, we could not secure the time to extract the actual song information like artist name or genre name on different platforms and evaluate GTZAN ourselves, but according to Sturm, these figures exhibit the real data, collected from last.fm. The last.fm provides a "count" for each one: a normalized quantity such that 100 means the tag is applied by most listeners, and 0 means the tag is applied by the fewest. They kept only those tags with counts greater than 0. The fourth column of Table 1 shows the number of tracks we identify that have tags in last.fm, and the number of tags with non-zero count. When they could not find tags for a song, they got the tags applied to the artist. For instance, though they identified all 100 excerpts in Blues, only 75 of the songs are tagged. Of these, they got 2,904 tags with non-zero counts. For the remaining 25 songs, they retrieved 2,061 tags from the tags given to the artists.

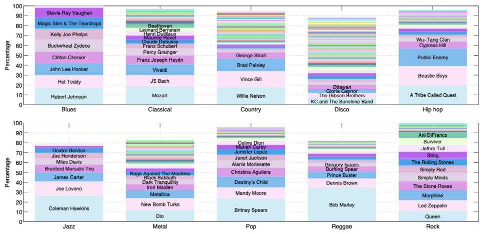


Figure 1: Artist composition of each *GTZAN* category. We do not include unidentified excerpts.

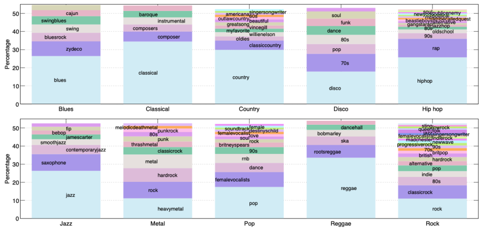


Figure 2: Top tags of each *GTZAN* category. We do not include unidentified excerpts.

Precedent and Reliability: GTZAN has been widely used in previous research, making it a reliable benchmark for music genre classification. The dataset's balanced representation across multiple genres makes it ideal for comparative studies, and its popularity in academic literature allows for performance benchmarking against well-established models. Additionally, the dataset provides consistent quality and length across tracks, which helps in maintaining homogeneity during feature extraction.

3.1.2 Preprocessing Steps

Loading Audio Files: We used Librosa to load the audio files at a consistent sampling rate of 22,050 Hz, ensuring uniformity across all tracks. This consistent rate allows for comparable feature extraction across different audio samples. Jazz.00054.wav failed to be processed correctly due to potential corruption or unreadable data. As a result, rows corresponding to these problematic audio files were skipped to avoid introducing noise or errors into the dataset.

Feature Extraction: To enhance the performance of the genre classification task, we extracted multiple types of features from the audio files,

Mel Spectrogram: Mel spectrograms were computed using Librosa to convert the audio signals into a time-frequency representation, capturing the distribution of energy across different frequency bands. These spectrograms provide a comprehensive overview of the harmonic and rhythmic characteristics of the audio.

Wavelet Transforms: Wavelet transformations were applied using PyWavelets to extract detailed temporal features from the audio signals. This approach helps to capture sudden changes and transient information that is characteristic of different music genres.

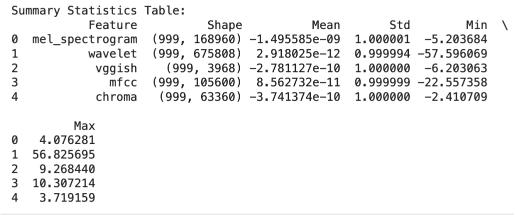
MFCC Features: Mel-Frequency Cepstral Coefficients (MFCC) were extracted using Librosa, with a configuration of 40 coefficients per frame, capturing the timbral information of the audio. MFCCs are widely used in speech and music analysis to represent spectral properties in a compact way.

Chroma Features: Chroma features were extracted to capture harmonic and pitch-related information. They were computed using Librosa, with a configuration of 24 chroma bins, which helped capture the tonal content of the audio. This feature is particularly useful for distinguishing between genres based on harmonic structures and chord progressions.

VGGish Embeddings: To leverage pre-trained embeddings from a large-scale audio dataset, we extracted features using Google's VGGish model. These embeddings provided a high-level representation of the audio, capturing both temporal and spectral characteristics of the music.

Feature Unification: After extracting the features, the length of each feature type was unified. We determined the maximum length among all samples for each feature type, then padded shorter samples to match this maximum length. This padding ensured that the inputs had consistent shapes for deep learning models.

Standardization: The extracted features were standardized using StandardScaler from sklearn to normalize the data. Each feature type was scaled independently, ensuring that they all have a mean of zero and unit variance. This step helps improve the convergence rate during model training by normalizing different feature scales.



```
Summary Statistics Table:
           Feature          Shape          Mean       Std        Min  \
0  mel_spectrogram  (999, 168960) -1.495585e-09  1.000001  -5.203684
1          wavelet  (999, 675808)  2.918025e-12  0.999994 -57.596069
2           vggish    (999, 3968) -2.781127e-10  1.000000  -6.203063
3             mfcc  (999, 105600)  8.562732e-11  0.999999 -22.557358
4           chroma   (999, 63360) -3.741374e-10  1.000000  -2.410709

         Max
0   4.076281
1  56.825695
2   9.268440
3  10.307214
4   3.719159
```

Figure 2: The summary statistics report for standardized features

Data Splitting: The dataset was split into training, validation, and testing sets using a 60-20-20 split ratio with train_test_split from sklearn. Stratified sampling was employed to maintain a balanced distribution of genres across all subsets, ensuring effective and unbiased model evaluation.

Padding for CUDNN Compatibility: To ensure compatibility with CUDNN, we further padded the features to reduce the input length, with a maximum length set based on the 70th percentile of the feature lengths, but capped at a minimum of 15,000. This step helps to control the computational cost while keeping the most significant portion of the data. By setting a minimum threshold, we ensured that enough temporal context was preserved for effective learning by the model.

Final Input Preparation: The features were reshaped to add an additional channel dimension, making them suitable for convolutional and recurrent neural networks. This ensured that the input data format matched the expected input dimensions for deep learning layers, such as Conv2D and LSTM.

3.2 ML Approaches

3.2.1 *Deep Learning Models*

We conducted a series of experiments to evaluate the performance of deep learning models for music genre classification, using different combinations of audio features.

Convolutional Neural Networks (CNN): The CNN model was trained using Mel spectrograms and chroma features, enabling it to effectively capture spatial hierarchies in the time-frequency domain. These hierarchies are essential for identifying genre-specific harmonic and rhythmic patterns.

Recurrent Neural Networks (RNN): The RNN model, specifically utilizing Long Short-Term Memory (LSTM) networks, was trained on wavelet and MFCC features. This approach allowed the model to capture temporal dependencies in audio sequences, effectively representing the sequential nature of music.

Hybrid CNN-RNN Models: The hybrid model combined the spatial feature extraction of the CNN with the temporal modeling capabilities of LSTMs. This model used Mel spectrograms, chroma, wavelet, MFCC, and VGGish features, enabling a comprehensive representation of both spatial and sequential aspects of the music.

Transfer Learning: The hybrid model also incorporated transfer learning by leveraging the pre-trained VGGish model for feature extraction. This approach allowed the model to utilize high-level audio features that had already been trained on a broad range of sounds, aiding in improved genre classification performance.

3.2.2 *Baseline Models and Benchmark Methods*

To evaluate the effectiveness of our deep learning models, we established several baseline models for comparison, which used Mel spectrograms, chroma, wavelet, and MFCC features extracted from the audio data (excluding VGGish).

Logistic Regression: Logistic Regression served as a benchmark to evaluate how well the handcrafted features could be linearly classified.

Stochastic Gradient Descent (SGD): SGD provided a comparison to understand the performance difference between simpler linear models and more complex deep learning models. SGD optimization allowed us to efficiently train a linear classifier using the extracted audio features.

Random Forest: The Random Forest model, ss an ensemble method, combined multiple decision trees to capture non-linear relationships, serving as a more robust benchmark compared to the linear models. This model aimed to highlight the difference in generalizability between feature-engineered models and deep learning models.

3.3 Evaluation

To evaluate the success of our models, we will use the following metrics

3.3.1 *Deep Learning Models* :

Accuracy & Loss Over Epochs: We use accuracy and loss curves to visualize both training and validation performance over epochs. This helps to assess convergence behavior, overfitting, or underfitting risks throughout the training process.

Heatmap for Predictions: A prediction heatmap is generated to evaluate the model's performance on the test set. The heatmap illustrates the prediction distribution across classes, highlighting misclassifications and prediction trends.

3.3.2 *Baseline Models* :

Accuracy Score: The accuracy score is used to evaluate the model performance on the validation and test sets. This serves as a key metric to compare the baseline models with the deep learning approaches.

Classification Report: A classification report is generated, providing metrics such as precision, recall, and F1-score for each class. This allows a detailed assessment of model performance across all classes and helps to directly compare the classification ability of different models.

4 Result

4.1 CNN

The CNN architecture leverages Conv2D layers, followed by batch normalization, dropout, and residual connections. These components work in synergy to enhance feature learning and improve model robustness, thereby contributing to the performance observed in both training and validation metrics.

Figure 3 illustrates the training and validation performance of the CNN model across 250 epochs. The model employs a learning rate warm-up strategy during the initial 10 epochs, followed by a cosine annealing schedule for the remaining training period. This learning rate schedule helps in achieving smoother convergence—initially by enabling exploration through a higher learning rate, and later by gradually reducing the learning rate to focus on promising areas in the parameter space. The upward trend in both training and validation accuracies suggests effective feature extraction, though the validation accuracy plateaus around 0.6, highlighting possible overfitting or the need for additional regularization.

The training loss steadily declines, whereas the validation loss fluctuates in response to the cosine annealing schedule. The dynamic adjustment of learning rates likely aids in escaping local minima but could also contribute to the observed variability in validation metrics. Introducing early stopping or increasing regularization strength might address this fluctuation and improve generalization.

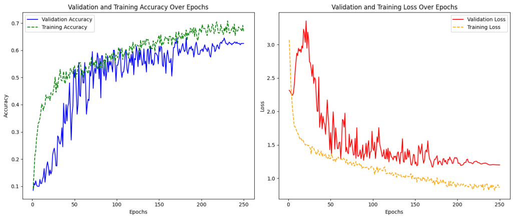
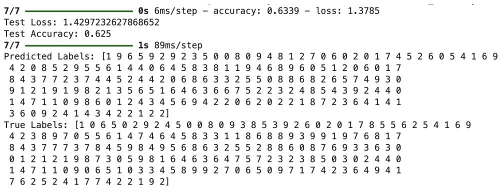


Figure 3: CNN Accuracy & Loss Over Epochs

Figure 4 shows the final test predictions of the model, achieving an overall accuracy of 62.5%. This indicates the model's moderate ability to learn genre-specific features. However, challenges remain, particularly with genres that have overlapping auditory characteristics, underscoring the need for further optimization of model parameters or feature representation strategies.



```
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.6339 - loss: 1.3785
Test Loss: 1.4297232627868652
Test Accuracy: 0.625
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 89ms/step
Predicted Labels: [1 9 6 5 9 2 9 2 3 5 0 0 8 0 9 4 8 1 2 7 0 6 0 2 0 1 7 4 5 2 6 0 5 4 1 6 9
 4 2 0 8 5 2 9 5 5 6 1 4 4 0 6 4 5 8 3 8 1 1 9 4 6 8 9 6 0 5 1 2 0 6 0 1 7
 8 4 3 7 7 2 3 7 4 4 5 2 4 4 2 0 6 8 6 3 3 2 5 5 0 8 8 6 8 2 6 5 7 4 9 3 0
 9 1 2 1 9 1 9 8 2 1 3 5 6 5 1 6 4 6 3 6 6 7 5 2 2 3 2 4 8 5 4 3 9 2 4 4 0
 1 4 7 1 1 0 9 8 6 0 1 2 4 3 4 5 6 9 4 2 2 0 6 2 0 2 2 1 8 7 2 3 6 4 1 4 1
 3 6 0 9 2 4 1 4 3 4 2 2 1 2 2]
True Labels: [1 0 6 5 0 2 9 2 4 5 0 0 8 0 9 3 8 5 3 9 2 6 0 2 0 1 7 8 5 5 6 2 5 4 1 6 9
 4 2 3 8 9 7 0 5 5 6 1 4 7 4 6 4 5 8 3 3 1 1 8 6 8 8 9 3 9 9 1 9 7 6 8 1 7
 8 4 3 7 7 7 3 7 8 4 5 9 8 4 9 5 6 8 6 3 2 5 5 2 8 8 6 0 8 7 6 9 3 3 6 3 0
 0 1 2 1 2 1 9 8 7 3 0 5 9 8 1 6 4 6 3 6 4 7 5 7 2 3 2 3 8 5 0 3 0 2 4 4 0
 1 4 7 1 1 0 9 0 6 5 1 0 3 3 4 5 8 9 9 2 7 0 6 5 0 9 7 1 7 4 2 3 6 4 9 4 1
 7 6 2 5 2 4 1 7 7 4 2 2 1 9 2]
```

Figure 4: CNN final prediction

The heatmap in Figure 5 presents the confusion matrix of predictions for various genres. The model distinguishes genres like classical and metal with high accuracy, as reflected by the darker diagonal elements. Nevertheless, it encounters difficulties when predicting genres such as pop and disco, which share similar tonal features. The use of the cosine annealing learning rate schedule has facilitated learning genre-specific features, but there is room for improvement in achieving consistent differentiation among genres with overlapping characteristics. Future efforts could explore richer feature representations or multi-task learning approaches to enhance this differentiation.

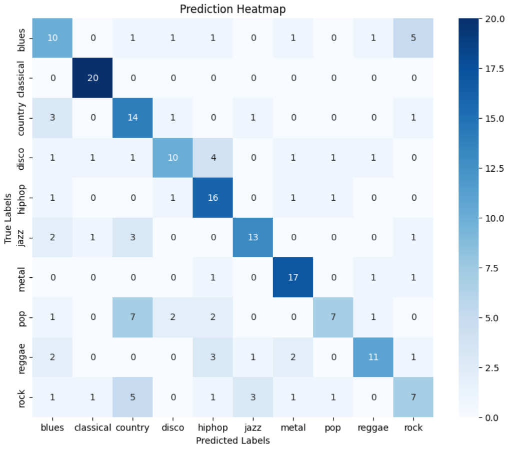


Figure 5: CNN Heatmap for Predictions

4.2 RNN (LSTM)

Figure 6 presents the performance of the RNN model for genre classification over 200 epochs. The RNN model uses bidirectional LSTMs to capture temporal dependencies. Both training and validation accuracies gradually improve, though a significant gap emerges between the two curves, suggesting overfitting.

For the loss behavior (Figure 6), training loss reduces smoothly, whereas validation loss fluctuates heavily, even after cosine annealing. Despite the gradual reduction in the validation loss, the difference between training and validation losses implies that the RNN may not generalize as well to unseen data.

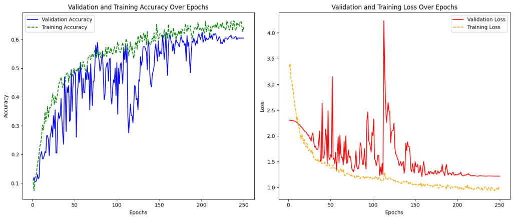
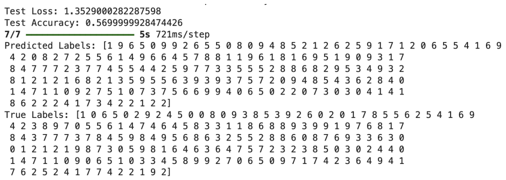


Figure 6: RNN Accuracy & Loss Over Epochs



```
Test Loss: 1.3529000282287598
Test Accuracy: 0.5699999928474426
7/7 ━━━━━━━━━━━━━━━━━━━━ 5s 721ms/step
Predicted Labels: [1 9 6 5 0 9 9 2 6 5 5 0 8 0 9 4 8 5 2 1 2 6 2 5 9 1 7 1 2 0 6 5 5 4 1 6 9
 4 2 0 8 2 7 2 5 5 6 1 4 9 6 6 4 5 7 8 8 1 1 9 6 1 8 1 6 9 5 1 9 0 9 3 1 7
 8 4 7 7 7 2 3 7 7 4 5 5 4 4 2 5 9 7 7 3 3 5 5 5 2 8 8 6 8 2 9 5 3 4 9 3 2
 8 1 2 1 2 1 6 8 2 1 3 5 9 5 5 6 3 9 3 9 3 7 5 7 2 0 9 4 8 5 4 3 6 2 8 4 0
 1 4 7 1 1 0 9 2 7 5 1 0 7 3 7 5 6 6 9 9 4 0 6 5 0 2 2 0 7 3 0 3 0 4 1 4 1
 8 6 2 2 2 4 1 7 3 4 2 2 1 2 2]
True Labels: [1 0 6 5 0 2 9 2 4 5 0 0 8 0 9 3 8 5 3 9 2 6 0 2 0 1 7 8 5 5 6 2 5 4 1 6 9
 4 2 3 8 9 7 0 5 5 6 1 4 7 4 6 4 5 8 3 3 1 1 8 6 8 8 9 3 9 9 1 9 7 6 8 1 7
 8 4 3 7 7 7 3 7 8 4 5 9 8 4 9 5 6 8 6 3 2 5 5 2 8 8 6 0 8 7 6 9 3 3 6 3 0
 0 1 2 1 2 1 9 8 7 3 0 5 9 8 1 6 4 6 3 6 4 7 5 7 2 3 2 3 8 5 0 3 0 2 4 4 0
 1 4 7 1 1 0 9 0 6 5 1 0 3 3 4 5 8 9 9 2 7 0 6 5 0 9 7 1 7 4 2 3 6 4 9 4 1
 7 6 2 5 2 4 1 7 7 4 2 2 1 9 2]
```

Figure 7: RNN final prediction

The RNN achieved a test accuracy of about 0.57, as shown in Figure 7. A closer look at the prediction heatmap (Figure 8) highlights that certain genres like "metal" and "classical" had fewer misclassifications, similar to the CNN model, whereas others, including "jazz" and "disco," faced more frequent confusion. The challenges may be attributed to subtle temporal features that require more refined tuning or additional regularization

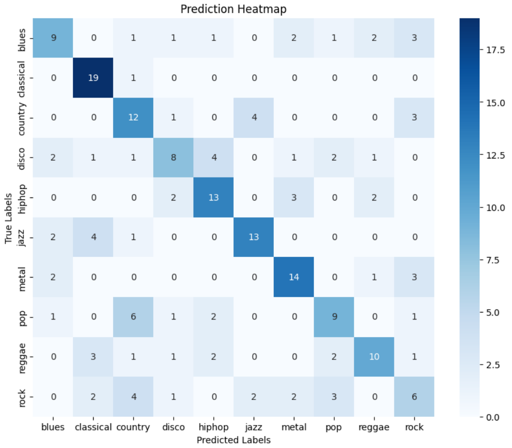


Figure 8: RNN Heatmap for Predictions

4.3 CNN-RNN Hybrid with Transfer Learning

Our hybrid model integrates the strengths of CNN and RNN components and incorporates transfer learning using pre-trained "VGGish" embeddings for feature extraction. The model processed with all other features through convolutional and recurrent architectures, enabling the effective combination of spatial, temporal, and pre-learned features.

The training and validation curves for the hybrid model are shown in Figure 9. The accuracy and loss plots demonstrate a consistent improvement over the epochs, with validation accuracy stabilizing after approximately 200 epochs.

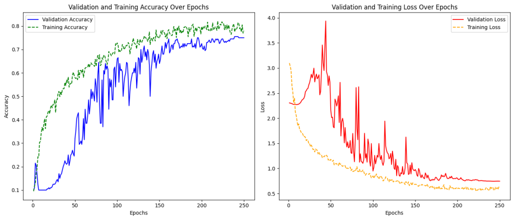
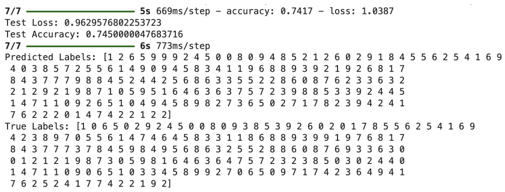


Figure 9: Hybrid Accuracy & Loss Over Epochs



```
7/7 ━━━━━━━━━━━━━━━━━━━━ 5s 669ms/step - accuracy: 0.7417 - loss: 1.0387
Test Loss: 0.9629576802253723
Test Accuracy: 0.7450000047683716
7/7 ━━━━━━━━━━━━━━━━━━━━ 6s 773ms/step
Predicted Labels: [1 2 6 5 9 9 9 2 4 5 0 0 8 0 9 4 8 5 2 1 2 6 0 2 9 1 8 4 5 5 6 2 5 4 1 6 9
 4 0 3 8 5 7 2 5 5 6 1 4 9 0 9 4 5 8 3 4 1 1 9 6 8 8 9 3 9 2 1 9 2 6 8 1 7
 8 4 3 7 7 7 9 8 8 4 5 2 4 4 2 5 6 8 6 3 3 5 5 2 2 8 6 0 8 7 6 2 3 3 6 3 2
 2 1 2 9 2 1 9 8 7 1 0 5 9 5 1 6 4 6 3 6 3 7 5 7 2 3 9 8 8 5 3 3 9 2 4 4 5
 1 4 7 1 1 0 9 2 6 5 1 0 4 9 4 5 8 9 8 2 7 3 6 5 0 2 7 1 7 8 2 3 9 4 2 4 1
 7 6 2 2 2 0 1 4 7 4 2 2 1 2 2]
True Labels: [1 0 6 5 0 2 9 2 4 5 0 0 8 0 9 3 8 5 3 9 2 6 0 2 0 1 7 8 5 5 6 2 5 4 1 6 9
 4 2 3 8 9 7 0 5 5 6 1 4 7 4 6 4 5 8 3 3 1 1 8 6 8 8 9 3 9 9 1 9 7 6 8 1 7
 8 4 3 7 7 7 3 7 8 4 5 9 8 4 9 5 6 8 6 3 2 5 5 2 8 8 6 0 8 7 6 9 3 3 6 3 0
 0 1 2 1 2 1 9 8 7 3 0 5 9 8 1 6 4 6 3 6 4 7 5 7 2 3 2 3 8 5 0 3 0 2 4 4 0
 1 4 7 1 1 0 9 0 6 5 1 0 3 3 4 5 8 9 9 2 7 0 6 5 0 9 7 1 7 4 2 3 6 4 9 4 1
 7 6 2 5 2 4 1 7 7 4 2 2 1 9 2]
```

Figure 10: Hybrid final prediction

In Figure 10, the model's final predictions are compared to the true labels on the test set, achieving a test accuracy of 0.745. The heatmap in Figure 11 further illustrates that the hybrid model maintains higher consistency across different genres compared to the CNN or RNN models, indicating better generalizability owing to the diverse feature set and the use of transfer learning. However, the hybrid model still struggles with certain genres, particularly jazz and country, which are more challenging due to their nuanced characteristics and overlapping elements with other genres like blues and folk. This suggests that while the hybrid model benefits from combining both spatial and temporal features, further optimization is needed to accurately capture subtle genre-specific variations in these more complex and overlapping categories.

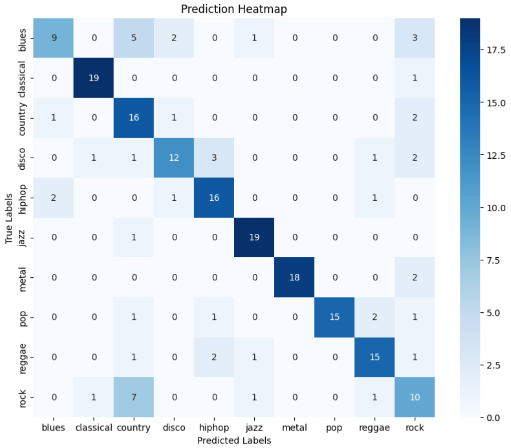
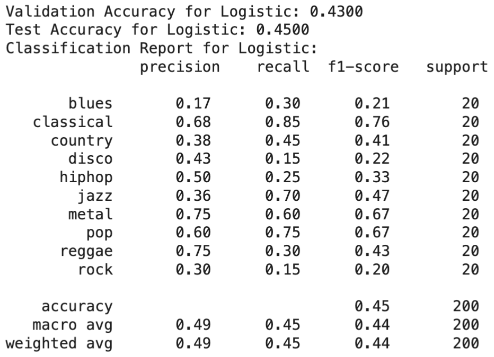


Figure 11: Hybrid Heatmap for Predictions

4.4 Logistic Regression

Figure 12 shows that the overall test accuracy for the logistic model is 0.45, which is considerably lower than that of the deep learning models. The F1-scores vary significantly across genres, with classical music achieving the highest F1-score of 0.76, while blues and rock have notably lower scores of 0.21 and 0.20, respectively.

Compared to deep learning models, logistic regression has limited generalization, especially for complex genres, due to its reliance on handcrafted features and a simple linear decision boundary. In contrast, the CNN, RNN, and hybrid models can capture nuanced temporal and harmonic relationships, highlighting the advantage of richer feature representations and more sophisticated modeling.



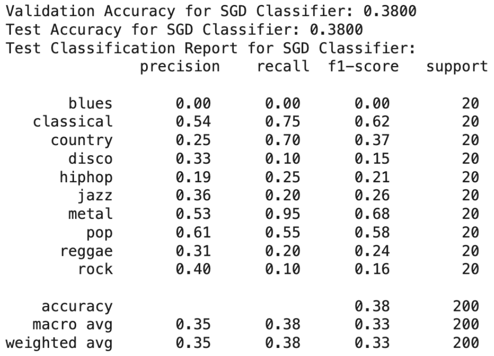
```
Validation Accuracy for Logistic: 0.4300
Test Accuracy for Logistic: 0.4500
Classification Report for Logistic:
               precision    recall  f1-score   support

       blues       0.17      0.30      0.21        20
   classical       0.68      0.85      0.76        20
     country       0.38      0.45      0.41        20
       disco       0.43      0.15      0.22        20
      hiphop       0.50      0.25      0.33        20
        jazz       0.36      0.70      0.47        20
       metal       0.75      0.60      0.67        20
         pop       0.60      0.75      0.67        20
      reggae       0.75      0.30      0.43        20
        rock       0.30      0.15      0.20        20

    accuracy                           0.45       200
   macro avg       0.49      0.45      0.44       200
weighted avg       0.49      0.45      0.44       200
```

Figure 12: Logistic Regression Classification report

4.5 Stochastic Gradient Descent (SGD)

The SGD model achieved a test accuracy of 0.38. It performed reasonably well for classical and metal genres, with F1-scores of 0.62 and 0.68, but struggled with blues and disco, achieving nearly zero scores.

Compared to deep learning models, SGD lacks the ability to learn complex interactions and temporal relationships, relying instead on simple linear separations. This limitation results in overall poorer performance, particularly for genres with nuanced differences. These findings show that while SGD provides a basic benchmark, it falls short of the generalization capability of deep learning models.



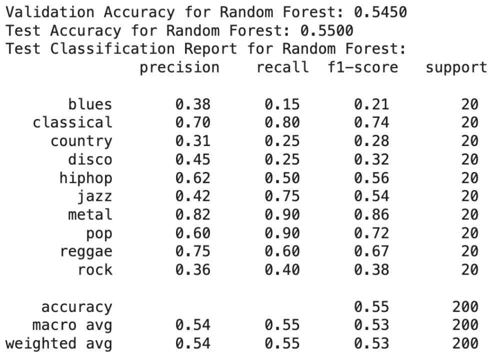
```
Validation Accuracy for SGD Classifier: 0.3800
Test Accuracy for SGD Classifier: 0.3800
Test Classification Report for SGD Classifier:
               precision    recall  f1-score   support

       blues       0.00      0.00      0.00        20
   classical       0.54      0.75      0.62        20
     country       0.25      0.70      0.37        20
       disco       0.33      0.10      0.15        20
      hiphop       0.19      0.25      0.21        20
        jazz       0.36      0.20      0.26        20
       metal       0.53      0.95      0.68        20
         pop       0.61      0.55      0.58        20
      reggae       0.31      0.20      0.24        20
        rock       0.40      0.10      0.16        20

    accuracy                           0.38       200
   macro avg       0.35      0.38      0.33       200
weighted avg       0.35      0.38      0.33       200
```

Figure 13: SGD Classification report

4.6 Random Forest

Figure 14 shows that the Random Forest model achieved a test accuracy of 0.55, an improvement over logistic regression and SGD. The model performed particularly well for classical and metal genres, achieving F1-scores of 0.74 and 0.86, respectively, but struggled with blues and country, with F1-scores of only 0.21 and 0.28.

Compared to deep learning models, Random Forest is less capable of capturing temporal and sequential features inherent in audio data, which is especially evident in genres like blues and jazz. Despite improvements over simpler baselines, its generalizability is still limited. Interestingly, when using VGGish features, the accuracy of the Random Forest increased significantly, reaching 80%, highlighting the impact of pre-trained deep features even with traditional models.



```
Validation Accuracy for Random Forest: 0.5450
Test Accuracy for Random Forest: 0.5500
Test Classification Report for Random Forest:
               precision    recall  f1-score   support

       blues       0.38      0.15      0.21        20
   classical       0.70      0.80      0.74        20
     country       0.31      0.25      0.28        20
       disco       0.45      0.25      0.32        20
      hiphop       0.62      0.50      0.56        20
        jazz       0.42      0.75      0.54        20
       metal       0.82      0.90      0.86        20
         pop       0.60      0.90      0.72        20
      reggae       0.75      0.60      0.67        20
        rock       0.36      0.40      0.38        20

    accuracy                           0.55       200
   macro avg       0.54      0.55      0.53       200
weighted avg       0.54      0.55      0.53       200
```

Figure 14: Random Forest Classification report

5 Discussion

In this project, we primarily focused on quantifiable audio features such as Mel-frequency, chroma, and wavelet transformations to assess the capability of deep learning models and traditional baselines in genre classification. While these features provide insight into the temporal and harmonic elements of music, they do not fully encapsulate the complexity of musical genres as experienced by human listeners. In real-world contexts, distinguishing between genres is often subjective. For instance, in recent years, hip-hop has increasingly overlapped with pop music, and the definition of pop itself has become fluid, with its characteristics changing across different eras. This evolving nature of genre boundaries suggests that deep learning models need continuous fine-tuning to reflect real-time trends effectively.

An alternative approach, such as the one employed by music streaming platforms like Spotify, goes beyond analyzing just the audio waveform. Spotify incorporates more emotional features like danceability, valence, and energy to understand user preferences. These emotional attributes are extracted not only from audio analysis but also by using collaborative filtering to track user behavior and preferences, thus defining genres based on user interactions. By leveraging user statistics, Spotify can adapt more effectively to changing genre definitions, while reinforcing user engagement through personalized recommendations. This approach highlights an important limitation of our deep learning models—though effective in capturing structured audio features, they do not inherently account for the evolving and subjective nature of music genres, which require additional context and user feedback for accurate classification.

6 Limitation

Our project utilized the GTZAN dataset, which, as noted by Sturm, has several significant shortcomings that affect the reliability of genre classification models. One major issue is data repetition; the dataset contains 50 exact repetitions and 21 recording repetitions, where the same song appears in slightly different versions. Additionally, the dataset suffers from artist repetition—100 excerpts in the "Blues" category come from only nine artists, and more than a third of "Reggae" songs are by Bob Marley. This lack of variety limits the generalizability of models trained on GTZAN, as they may become biased towards specific artists rather than the broader characteristics of a genre. Classical music also includes different orchestras playing the same compositions, further contributing to this lack of diversity.

Mislabeling is another significant problem in the GTZAN dataset. By comparing GTZAN with other user-labeled datasets such as Last.fm, many instances of mislabeled genres have been found. For example, the song "Leaning On The Everlasting Arm" by Ladysmith Black Mambazo is labeled as "Pop" in GTZAN, whereas fan-labeled tags on Last.fm suggest it is better categorized as "African" or "World." Such mislabeling introduces noise and reduces the accuracy of machine learning models by making it more challenging to learn correct genre distinctions.

Finally, the dataset contains several distorted audio files, further hindering its usability. For instance, the "Reggae 86" file is so heavily distorted that its final 25 seconds are practically unusable. These distortions impact the model's ability to extract meaningful features from the audio.

To address these issues and achieve cleaner machine learning results, it is necessary to either clean and normalize the GTZAN dataset thoroughly or opt for a more reliable audio dataset with greater variety and accurate labels. Ensuring that the training data is free of repetition, mislabeling, and distortions is crucial for building models capable of accurate and generalizable genre classification.

7 Conclusion

In conclusion, we explored a range of machine learning approaches, including both traditional baseline models and advanced deep learning techniques, for music genre classification. Our experiments included Logistic Regression, SGD, and Random Forest as baseline models, along with CNN, RNN, hybrid models, and transfer learning using VGGish. The results showed that deep learning models consistently outperformed the baselines by capturing more complex temporal and harmonic features from audio data.

The CNN and RNN models captured distinct spatial and temporal characteristics, respectively, while the hybrid model effectively combined these strengths to achieve better generalization. Transfer learning, leveraging pre-trained features, further improved performance, particularly for more challenging genres. Although baseline models provided a reference point, they struggled with more nuanced genres, lacking the capacity to model intricate relationships within the data.

Nevertheless, our models faced limitations due to the well-documented issues in the GTZAN dataset, including duplicate samples, artist repetition, and mislabeled tracks, which hindered generalizability. Despite these challenges, our deep learning models—particularly the hybrid with transfer learning—showed substantial improvements over traditional methods.

For future work, we plan to address these dataset limitations by utilizing cleaner and more diverse audio datasets. Furthermore, we will explore integrating emotional features such as danceability and valence, akin to approaches used by platforms like Spotify. Including these features may provide richer insights into genre differentiation, especially when combined with listener feedback for more subjective classification.

Additionally, we aim to experiment with more advanced architectures, such as transformers, which excel at modeling long-term dependencies. The combination of transformers with CNN and RNN architectures has the potential to capture both local and global features effectively, offering a comprehensive representation of music's intricate spatial and temporal dynamics. These future improvements are intended to create a more robust, adaptive deep learning solution for music genre classification that aligns with evolving musical trends.

References

Sawan Rai. 2021. Music Genres Classification using Deep Learning Techniques. Data Science Blogathon, Analytics Vidhya. Retrieved June 2021 from https://www.analyticsvidhya.com/blog/2021/06/music-genres-classification-using-deep-learning-techniques/.

P. Tibadiya, "Music Genre Classification using RNN and LSTM," Medium, Apr. 17, 2020. [Online]. Available: https://medium.com/@premtibadiya/mu genre-classification-using-rnn-lstm-1c212ba21e06. [Accessed: Oct. 30, 2024].

Ruoho Ruotsi, LSTM-Music-Genre-Classification, GitHub, 2016 https://github.com/ruohoruotsi/LSTM-Music-Genre-Classification?tab=MIT-1-ov-file#readme

Sturm, Bob L. "The GTZAN Dataset: Its Contents, Its Faults, Their Effects on Evaluation, and Its Future Use." ArXiv (Cornell University), 6 June 2013, https://doi.org/10.48550/arxiv.1306.1461. Accessed 6 Nov. 2023.