DEEP LEARNING MODEL FOR MUSIC GENRE CLASSIFICATION USING CNN AND RNN IN PYTORCH

Winnie Hsiang
Student# 1008867852
winnie.hsiang@mail.utoronto.ca

Victor Deng
Student# 1008771615
victor.deng@mail.utoronto.ca

Coline Zhang
Student# 1008748027
coline.zhang@mail.utoronto.ca

Ashley Leal
Student# 1009122227
ashley.leal@mail.utoronto.ca

ABSTRACT

Music genre classification plays a pivotal role in enhancing the functionality of popular music streaming platforms, encompassing music recommendation systems, personalized playlists, and content organization. With the ever-expanding volume of music content available on the internet, the demand for efficient and automated music classification systems becomes increasingly vital. The following paper discusses a project for APS360 that performs music genre classification. This is achieved by utilizing a hybrid CNN and RNN model to analyze audio clip samples, predicting a genre from a selection of 10 possible classes.
—-Total Pages: 9

1 INTRODUCTION

Music is a universal language that attracts audiences from all over the world and plays a significant role in cultural development. With the rise of digital platforms and various social networking sites, there is an increasing demand for systems capable of organizing and categorizing music into different genres. Deep learning, especially when leveraging PyTorch, offers an ideal approach for this task, as it can learn from pre-classified music and extract crucial information for accurately classifying newly created tracks. These genre classification systems enable browsing websites or mobile apps to efficiently categorize recently uploaded songs and build a comprehensive database. In our project, our objective was to develop an efficient and effective music genre classification system utilizing state-of-the-art datasets and several modern deep learning techniques.

2 ILLUSTRATION / FIGURE

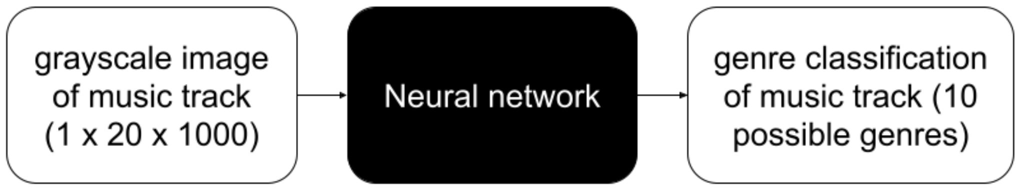

grayscale image of music track (1 x 20 x 1000)
Neural network
genre classification of music track (10 possible genres)

Figure 1: Blackbox Representation of Music Classification Model

3 Background and Related Work

To offer an overview of the advancements of music classification technology, we have selected six works in this domain. These publications have been chosen to aid us in accomplishing our project objectives.

Several key papers and works have contributed to the progress of CNN music classification models. One notable paper by Li et al (Li et al., 2010), introduced a novel approach to extracting musical pattern features using convolutional neural networks (CNNs), laying the foundation for subsequent models. Namrata Dutt's tutorial-style articles (Dutt, 2022) on music genre classification provided valuable insights into feature extraction and CNN architecture, while Wenlong Zhang's research article (Zhang, 2022) offered a deep learning-based classification method for music genres, reinforcing fundamental ideas and analysis techniques. Furthermore, Nikki Pelchat's thesis paper (Pelchat, 2021) implemented a CNN for music genre classification, offering a detailed methodology and assessment of viability. Moreover, the commercially available feature Spotify AI DJ, which utilizes user preferences and data analysis to recommend music, provides relevant insights into using multiple different types of input, including raw audio data, song titles, and lyrics, to generate accurate predictions. In addition, Mohsin Ashraf from University of Centrol Punjab has also proven the feasibility of using both RNN and CNN network in music genre classification (Ashraf et al., 2023). He mentions that,the proposed hybrid architecture of CNN and Bi-GRU (RNN variant) using Mel-spectrogram achieved the best accuracy at 89.30 percent, whereas the hybridization of CNN and LSTM (RNN variant) using MFCC achieved the best accuracy at 76.40 percent.

4 Data Processing

For this project, we collected data from two distinct sources: the GTZAN dataset (Tzanetakis & Cook, 2002) and the MFA dataset. (Javadi, 2021) The purpose was to create a comprehensive and diverse dataset by combining these sources. Our data cleaning process involved several steps, which are outlined below in detail:

4.1 Data Collection

The GTZAN dataset and the MFA dataset were obtained from their respective sources. The GTZAN dataset contains 1000 balanced data from 10 music genres which aligns with our project, while the MFA dataset provides a collection of audio tracks from 16 different genres.

4.2 Data Cleaning and Formatting

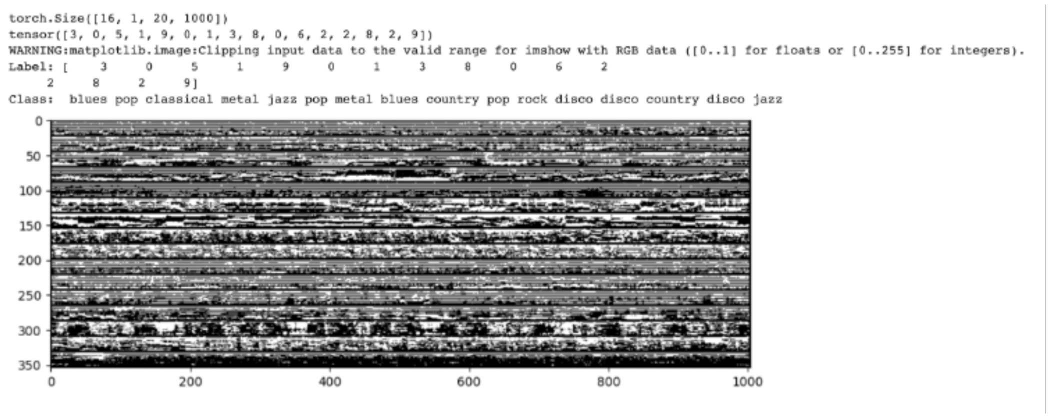
The data cleaning and formatting process involved the following steps:

1. Load all downloaded audio files along with ground-truth labels from local paths.
2. (for MFA dataset only) Format labels to align with project labels (e.g. 'Hip-Hop' into 'hiphop', 'Indie Rock' into 'rock')
3. (for MFA dataset only) Eliminate tracks with labels outside of the 10 categories.
4. Use the `librosa.mfcc` function to transform the audio clips into mel spectrograms.
5. Convert the resulting MFCCs into 2D numpy arrays for efficient handling and processing.
6. Set an infinite threshold when converting the arrays into strings to prevent data truncation.
7. Store the data in a pandas dataframe, with the MFCCs represented as strings along with their corresponding labels.
8. Save the dataframe as a complete `.csv` file for easy loading and use as input for the model.

4.3 Data Statistics and Examples

Our cleaned dataset consists of a total of 9,000 audio clips, each with a length of 30 seconds. The 9,000 samples were then split into a training set comprising 70%, a validation set consisting of 15%, and a test set encompassing 15%. To address the issue of unbalanced genres in the training and validation sets, we randomly sampled 5-second segments from minority classes until the desired data size was achieved. Ultimately, we obtained a balanced dataset of 100,000 training samples and 10,000 validation samples. As an illustrative example of a cleaned training sample, consider the following:

- Labels from the first batch of data and one sample melspectrum:



```
torch.Size([16, 1, 20, 1000])
tensor([3, 0, 5, 1, 9, 0, 1, 3, 8, 0, 6, 2, 2, 8, 2, 9])
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Label: [    3     0     5     1     9     0     1     3     8     0     6     2
     2     8     2     9]
Class:  blues pop classical metal jazz pop metal blues country pop rock disco disco country disco jazz
```

4.4 Plan for Final Testing Data

To evaluate the model's performance, we opted to divide the 30-second samples into six equal parts, resulting in a total of 7,428 5-second samples.

To assess the performance of our model on unseen data, we have collected `.mp3` or `.WAV` audio files with corresponding labels from different sources. These sources include additional datasets, personal playlists, and audio extraction from platforms like YouTube using crawlers. We aim to curate a final testing dataset with a minimum of 20 tracks for each genre category. Similar to the training data, the collected tracks will be segmented into non-overlapping 30-second segments and processed using the same techniques applied to the training data. This approach will provide us with a robust test set consisting of approximately 1,200 tracks, ensuring the model's evaluation on never-before-seen data.

5 Architecture

In our music genre classification project, our primary model consists total of 12 layers with 4 convolutional layers, 4 max pooling layers, single RNN layer, and 3 ANN layers for classification. We aim to extract relevant features from audio spectrograms and classify them into different genres using CRNN architecture.

During our training process, we utilized the CrossEntropyLoss as the chosen loss function along with the ReLU activation function. To optimize the model's weights, we employed the Adam optimizer. To enhance efficiency, GPU acceleration was harnessed, significantly reducing training time. To track progress and ensure reproducibility, we meticulously recorded training times and stored the model's parameters after each epoch for potential future use.

For monitoring the performance of our training and validation sets, we have implemented the `get_accuracy`, `get_loss`, and `get_accuracy_per_genre` functions. These functions monitor the accuracy and loss on both the training and validation sets. At the end of the training, we graph the accuracy and loss curves for better visualization of the model's performance.

5.1 Four CNN Layers with Four Maximum Pooling Layers

Beginning with a convolutional layer with 32 filters of size 5x5, each followed by ReLU activation, the process advances through max-pooling with 2x2 kernels. A second convolutional layer of 64 filters of the same size follows, again activated by ReLU and subsequently max-pooled. The sequence continues with a third convolutional layer of 128 filters sized 3x3, ReLU activation, and max-pooling. The final convolutional layer consists of 256 filters with the same 3x3 size, once more ReLU activated, and max-pooled.

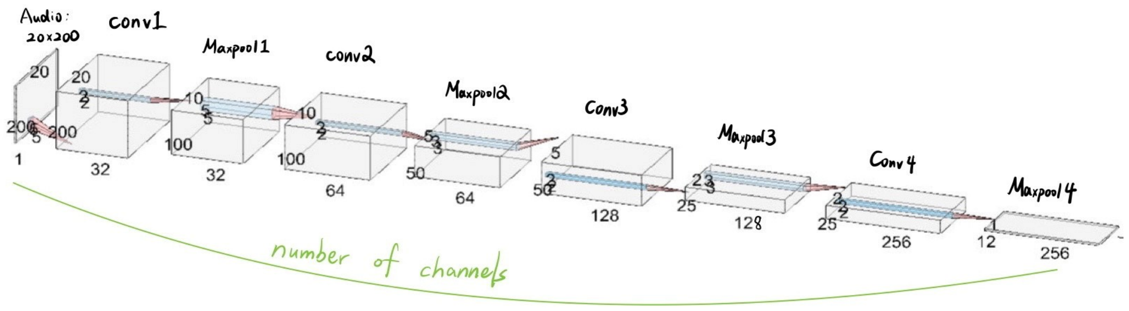


Figure 2: CNN Architecture

5.2 The RNN and ANN Layers

After the four CNN layers, we incorporate a bidirectional GRU layer with an input size of 256*1*12 = 3072 (256 channels times height and width) and a hidden size of 512. This RNN layer further processes the extracted features.

At the end of the model, we have three fully connected layers with linear transformations from size 1024 to 10, corresponding to the number of genres we aim to classify. We apply dropout after every linear layer to prevent overfitting. The softmax function is used within the CrossEntropyLoss criterion to scale numbers into probabilities for classifying the genres.

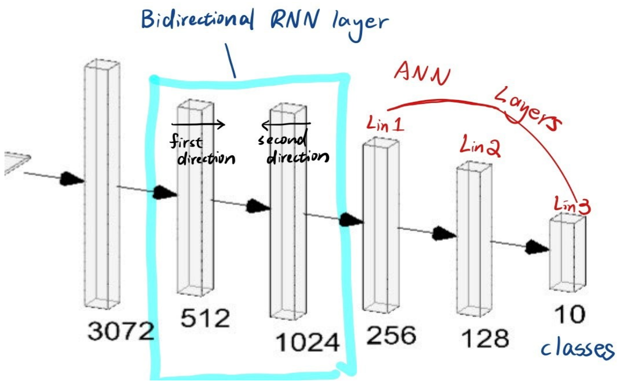


Figure 3: RNN and ANN Architecture

5.3 Training Processes and Final Hyperparameters

The following image demonstrates the final training and performance of the CRNN model. Our chosen hyperparameters for the final model are as follows:

- Learning rate: 0.00002
- Weight decay: 0.005
- Batch size: 200
- Number of epochs: 15

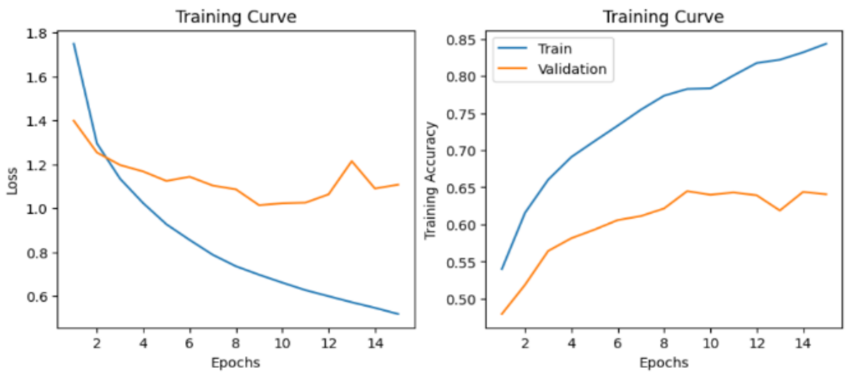


Figure 4: Final CRNN Model Performance

6 Baseline Model

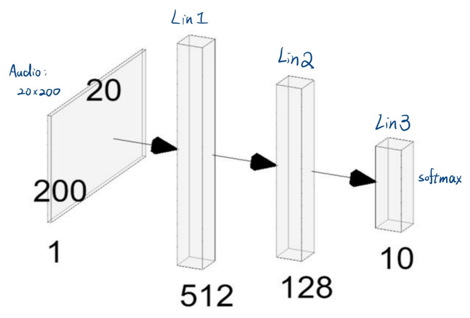
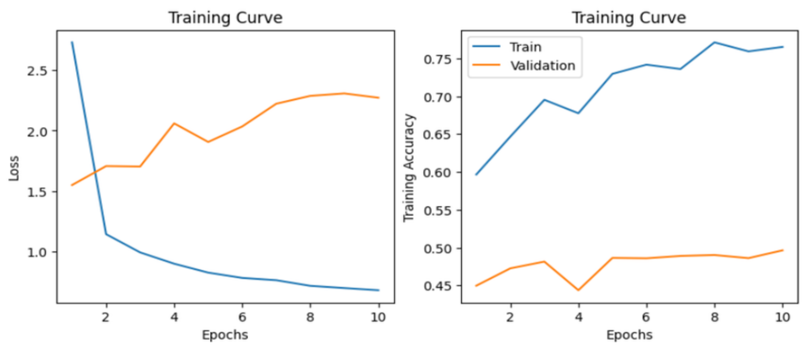


Figure 5: Baseline Model Architecture

In our baseline approach, audio spectrogram data was condensed into singular vectors. These vectors were then input into a three-layer fully connected neural network.

As a result, the model achieved a training accuracy of 77% and a validation accuracy of 50%. This performance indicates the model's potential to capture patterns in training data, though its ability to generalize to new data needs improvement.



```
Final Training Accuracy: 0.76508
Final Validation Accuracy: 0.4964
```

Figure 6: Baseline Model Performance

7 Quantitative Results

Our quantitative findings are satisfactory. Initially, we tested the accuracy of our model on two datasets with differing lengths. The first dataset consists of around 7000 pieces, each 5 seconds long, and equally distributed among 10 genres. With this dataset, our model achieved a testing accuracy of 66.5%, a satisfactory result. We then proceeded to test on 30-second pieces, using 16 songs from each genre. To accomplish this, we divided the 30-second pieces into six 5-second segments and asked the model to predict the genre for each 5-second part. We then assigned the most frequently predicted genre as the genre of the entire 30-second piece. By employing this method, our model benefited from a longer exposure to the audio clips and thus reached an accuracy of 70%, nearly matching human-level capability.

In addition, we utilized a confusion matrix to gain deeper insights into the model's performance across different genres. This confusion matrix was applied to the same 30-second music dataset described earlier and is illustrated in Figure 6. From the matrix, we can see that the model excels in identifying most genres, even achieving an impressive accuracy of 93.8% for both Classical and Metal genres. Conversely, its capability to classify Pop music is notably inadequate, with an accuracy rate of merely 31.2%. Notably, there is a significant chance (37.5%) that the model will misclassify a Pop song as Rock. While the confusion matrix showcases the model's overall strength and its exceptional ability to correctly predict genres such as Classical and Metal, it also reveals a specific area of weakness in the classification of Pop songs.

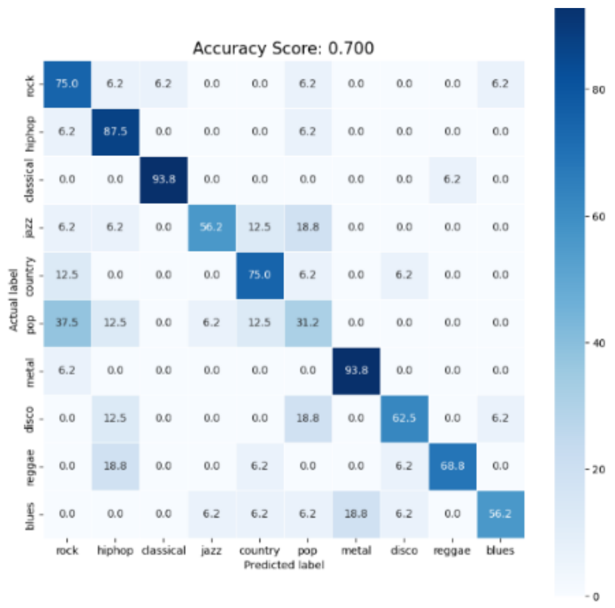


Figure 7: Confusion Matrix of Genre Classification

8 Qualitative Results

Our model's confusion matrix yielded unexpected results. Initially, we anticipated the genres of jazz and blues to be commonly confused due to their intertwined history and shared characteristics (Seagrove, 1915). Both utilize "blue" notes, swung notes, and syncopated rhythms, and predominantly employ the blues scale (Theory, 2023) (See blues scale). Jazz is often lively, swinging, and unpredictable, while blues exhibits a more melancholic and slow tempo (daw, 2023). Yet from an musical analytical standpoint, the unique harmonic and rhythmic attributes of jazz and blues set them apart in their own unique musical space, making them readily distinguishable from other genres.

However, the testing results revealed that the model adeptly differentiated between jazz and blues but frequently misclassified them with disparate genres like country and metal. This is shown in the confusion matrix in the above section. This suggests that our model excels at discerning tempo but

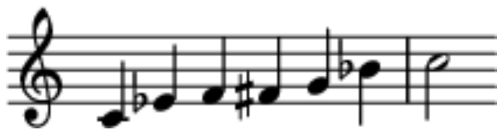
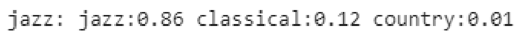


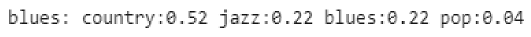
Figure 8: Blues Scale

struggles to identify specific harmonic progressions and musical attributes. This outcome can be attributed to the features of our input data. MFCCs capture the timbral texture and broad spectral shape of audio but are inherently limited in capturing melodic and harmonic nuances (Davis & Mermelstein, 1980).



```
jazz: jazz:0.86 classical:0.12 country:0.01
```

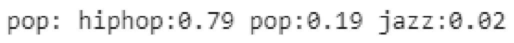
Figure 9: Sample Output for Jazz



```
blues: country:0.52 jazz:0.22 blues:0.22 pop:0.04
```

Figure 10: Sample Output for Blues

Another notable observation was the model's difficulty in classifying pop music, often mislabeling other genres under the pop category. The accuracy for this genre stood at 31.2%, with a false positive rate of 64.3%. This challenge might stem from the intrinsic nature of pop music. As a genre, pop borrows from numerous other musical styles and spans a wide range of tempos and characteristics, making it particularly challenging to define based on distinct attributes.



```
pop: hiphop:0.79 pop:0.19 jazz:0.02
```

Figure 11: Sample Output for a Pop Song

9 Evaluate model on new data

In order to assess the capabilities of our model in accurately identifying the genre of new music, we embarked on a comprehensive evaluation using new and unseen data. Initially, we utilized a test dataset obtained during the data processing step to gauge the model's performance. Subsequently, we applied the same evaluation method to a small dataset gathered from the Internet, enabling us to compare the model's genre-classification abilities with human aptitude.

We evaluated our model using a test dataset obtained during the data processing step, which was created by randomly dividing the original online music dataset into training, validation, and test subsets. We then formed a balanced 30-second test set with evenly distributed genres, as illustrated in the Figure 9 below. These new datasets were used to assess the model's performance, resulting in a test accuracy of 70%. This accuracy reflects the model's ability to classify music genres, considering the uniform length of the input data (30 seconds), the equal distribution of genres, and the separation of the test data to avoid premature exposure. This effort gave us valuable insights into the model's capabilities in identifying the genre of new music.

Furthermore, to simulate our model's real-world performance, we tested its ability using popular music found on YouTube, employing the same processing techniques as with our original training and validation data. We ensured these selections were not part of the original dataset and labeled each with the most recognized genre on YouTube. Through three separate tests, with a small dataset of 10 songs each, our model predicted the genres, achieving accuracies of 60%, 70%, and 60%. The consistent 60-70% accuracy range across these tests, as detailed in the confusion matrix from the first test shown in the Figure 10, confirms the model's robustness and aligns with our initial expectations in identifying new and unseen data.

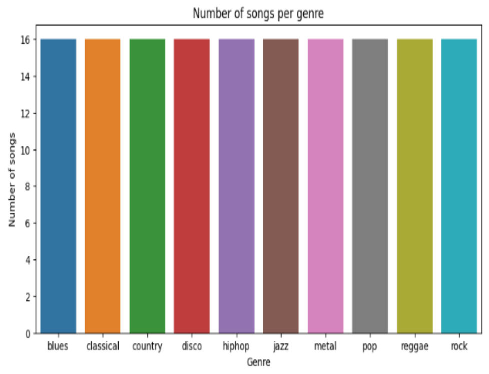
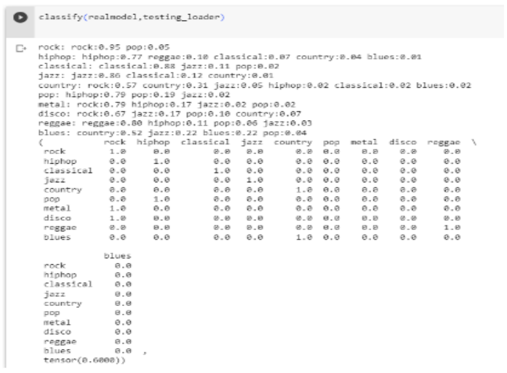


Figure 12: Genre Distribution in the Test Dataset



```
classify(realmodel,testing_loader)
```

```
rock: rock:0.95 pop:0.05
hiphop: hiphop:0.77 reggae:0.10 classical:0.07 country:0.04 blues:0.01
classical: classical:0.88 jazz:0.11 pop:0.02
jazz: jazz:0.86 classical:0.12 country:0.01
country: rock:0.57 country:0.31 jazz:0.05 hiphop:0.02 classical:0.02 blues:0.02
pop: hiphop:0.79 pop:0.19 jazz:0.02
metal: rock:0.79 hiphop:0.17 jazz:0.02 pop:0.02
disco: rock:0.67 jazz:0.17 pop:0.10 country:0.07
reggae: reggae:0.80 hiphop:0.11 pop:0.06 jazz:0.03
blues: country:0.52 jazz:0.22 blues:0.22 pop:0.04
(            rock  hiphop  classical  jazz  country  pop  metal  disco  reggae  \
 rock         1.0     0.0        0.0   0.0      0.0  0.0    0.0    0.0     0.0
 hiphop       0.0     1.0        0.0   0.0      0.0  0.0    0.0    0.0     0.0
 classical    0.0     0.0        1.0   0.0      0.0  0.0    0.0    0.0     0.0
 jazz         0.0     0.0        0.0   1.0      0.0  0.0    0.0    0.0     0.0
 country      0.0     0.0        0.0   0.0      1.0  0.0    0.0    0.0     0.0
 pop          0.0     1.0        0.0   0.0      0.0  0.0    0.0    0.0     0.0
 metal        1.0     0.0        0.0   0.0      0.0  0.0    0.0    0.0     0.0
 disco        1.0     0.0        0.0   0.0      0.0  0.0    0.0    0.0     0.0
 reggae       0.0     0.0        0.0   0.0      0.0  0.0    0.0    0.0     1.0
 blues        0.0     0.0        0.0   0.0      1.0  0.0    0.0    0.0     0.0

            blues
 rock         0.0
 hiphop       0.0
 classical    0.0
 jazz         0.0
 country      0.0
 pop          0.0
 metal        0.0
 disco        0.0
 reggae       0.0
 blues        0.0  ,
 tensor(0.6000))
```

Figure 13: Model's Performance on New Data

Finally, we juxtaposed the model's performance with human evaluations to gain more direct insights into its ability to classify genres for unseen data. During the three tests, we also invited two group members to manually identify the genres, resulting in scores of 65%, 65%, and 70% - almost identical to the model's performance. This reinforces our conclusion that the model achieves a human-level capability in music genre classification.

10 Discussion

Considering the large number of possible genre classes, the model's classification accuracy of 70% when tested on longer 30-second pieces, using a segmentation approach, and assigning the most frequently predicted genre is remarkable and even rivals human capabilities. This enhancement to the initial test configuration suggests that the model benefits from extended exposure to audio clips, allowing it to make more informed predictions.

The confusion matrix provides deeper insights into the model's performance across different genres. The matrix highlighted the model's strengths and weaknesses, showing an impressive accuracy of 93.8% for Classical and Metal genres but notable weaknesses in Pop songs with an accuracy of only 31.2%. This was an unexpected and noteworthy discrepancy.

Furthermore, qualitative analysis yielded some unexpected results. Jazz and blues, which share musical attributes, were surprisingly well-distinguished by the model. However, the model often struggled to classify them correctly and frequently misclassified them as disparate genres like Country and Metal. This suggests that while the model can discern tempo to a high degree, it encounters difficulties in capturing specific harmonic and melodic characteristics due to limitations in input data features.

In light of these findings, the model's performance is respectable, particularly given its highly accurate classifications of certain genres like Classical and Metal. However, there are areas of improvement, such as the frequent misclassifications of Pop, Jazz, and Blues. These results reveal the importance of considering the limitations of input data features and the complexities of certain music genres. Further refinement of the model's feature extraction and training on more diversified, representative, and balanced datasets could allow the model to yield more accurate genre classification results across all genres.

11 Ethical Considerations

Developing a music genre classification model requires careful attention to ethical considerations. This includes protecting user privacy, respecting copyright laws, promoting diversity and inclusivity, and acknowledging the limitations of genre classification. By addressing these concerns, we can ensure a responsible and inclusive approach to building the model.

11.1 Consent and Approval

Obtaining explicit consent from users before accessing their data is essential. Users should have full awareness of how their data will be used and have the opportunity to grant or deny permission. This transparent approach respects user autonomy and privacy rights.

11.2 Copyright Compliance

Training the model with copyrighted songs without proper authorization can lead to legal issues. It is important to obtain the necessary licenses or permissions from copyright holders to ensure compliance with copyright laws. This protects both the intellectual property rights of artists and avoids potential legal consequences.

11.3 Diversity and Inclusivity

Curating a diverse training dataset helps mitigate bias in the model's predictions. Including music from various genres, artists, and cultural backgrounds ensures that the model learns from a broad representation of music, reducing the risk of favoring certain genres or artists over others.

11.4 Limitations of Genre Classification

It is important to acknowledge and communicate the limitations of genre classification models to users. These models are designed to classify songs into predefined genres, and they may not accurately categorize songs that do not fit neatly into those genres. This caveat allows users to understand the model's scope and encourages them to consider additional context or options for songs that fall outside the predefined genres.

12 Project Difficulty and Quality

Considering the existence of numerous similar publicly available projects on the internet, along with the availability of pre-made datasets specifically designed for training a music genre classification model, the project's level of difficulty is reasonable for the amount of time given. The primary challenges faced by the team stemmed from hardware limitations during model training and the meticulous adjustment of hyperparameters to optimize both CNN and RNN architectures. Working with extensive datasets for model training, and constrained by Google Colab's slower GPUs and team members' personal devices for hardware, led to long training periods that hindered experimentation with different model configurations within a reasonable timeframe. Consequently, troubleshooting persisted over multiple days, requiring team members to invest extended periods of time. Given the complexity of our task and the project's allotted time, the achieved results exceeded initial expectations.

In this project, we extended beyond the knowledge gained in class, pushing ourselves to embrace concepts and techniques that were not covered in our curriculum. During the data processing phase, we employed MFCCs and Librosa to transform mp3 and wav files into MFCC diagrams. This approach allowed us to quantify our inputs and provided a visualization that enhanced our understanding of the data. Regarding the architectural choices, we ventured beyond the standard models covered in class. We innovatively combined them to form a CRNN model, aiming to leverage the strengths of both CNN and RNN architectures. This synergy aimed to deliver superior results. Lastly, for testing, our approach was multi-faceted. We evaluated our model from various angles and also drew comparisons with human judgement. This phase taught us the importance of establishing benchmarks and handling data meticulously to ensure the model wasn't prematurely exposed to any test data.

13 Link to Github or Colab Notebook

You can access the Colab notebook here.

https://colab.research.google.com/drive/1aYlNejVO8EdeaH0IZpNXDb610uQBQ-pq?usp=sharing

REFERENCES

What's the difference between jazz & blues?, 2023. URL `https://www.dawkes.co.uk/sound-room/whats-the-difference-between-jazz-blues/`.

Mohsin Ashraf, Fazeel Abid, Ikram Ud Din, Jawad Rasheed, Mirsat Yesiltepe, Sook Fern Yeo, and Merve T. Ersoy. A hybrid cnn and rnn variant model for music classification. *Applied Sciences*, 13(3):1476, 2023. doi: 10.3390/app13031476.

S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

Namrata Dutt. Music genre classification using cnn: Part 1- feature extraction. *MLearning.ai*, 1, 2022. URL `https://medium.com/mlearning-ai/music-genre-classification-using-cnn-part-1-feature-extraction-b417547b8981`.

S. Javadi. The many faces of anger: A multicultural video dataset of negative emotions in the wild (mfa-wild). 2021.

Tom Li, Antoni Chan, and Andy Chun. Automatic musical pattern feature extraction using convolutional neural network. *Lecture Notes in Engineering and Computer Science*, 2180, 03 2010.

Nikki Pelchat. Neural network music genre classification, 2021. Submitted to the Faculty of Graduate Studies and Research in Partial Fulfillment of the Requirements for the Degree of Master of Applied Science in Software Systems Engineering.

G. Seagrove. Blues is jazz and jazz is blues. *Chicago Daily Tribune*, 1915. Archived at Observatoire Musical Français, Paris-Sorbonne University. [Accessed Nov. 4, 2011].

Open Music Theory. Blues harmony, 2023. URL `https://viva.pressbooks.pub/openmusictheory/chapter/blues-harmony/`.

George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

Wenlong Zhang. Music genre classification based on deep learning. *Mobile Information Systems*, 2022:11, 2022. doi: 10.1155/2022/2376888.