CSE 4095/5095 - Final Project Report – Group 17

Matt Caro
University of Connecticut
Storrs, CT 06269
matthew.caro@uconn.edu

Ebube Jack-Davies
University of Connecticut
Storrs, CT 06269
ebube.jack-davies@uconn.edu

Emily Root
University of Connecticut
Storrs, CT 06269
emily.root@uconn.edu

Abstract

Music genre classification is an important task in music information retrieval (MIR). Traditional methods rely on hand-crafted features, but deep learning allows for automatic pattern recognition from visual representations of sounds. In this paper, we provide a convolutional neural network (CNN) model for genre classification that takes Mel spectrograms as input. We use the GTZAN dataset, which converts audio waveforms into spectrogram images for CNN processing. Our findings will show that deep learning can efficiently classify music genres based on spectrogram data.

1 Introduction

Spotify is one of the largest music streaming services in the world. It allows users to digitally stream songs from millions of different artists and musicians on virtually any device. With algorithms tailoring songs to each user's specific account, it allows for personalization that can not only keep the user engaged with music that they have already listened to but also expose them to new music they might enjoy. Music genre classification is undoubtedly a strong force that allows for this personalization to be accurate. While large streaming platforms already use AI in labelling genres, they rely on some data that is not directly found in the music itself. This includes information like the artist names, song titles, and features from the lyrics, like speechiness (the number of spoken words) or finding keywords specific to certain genres [1][2]. The goal of this study is to understand how deep learning can improve genre classification in different types of music by finding certain patterns and frequencies that would point to a certain type of music without relying on any external or manually labeled information.

2 Methodology

Working through this project consisted of four main parts: selecting and analyzing the dataset, preprocessing the selected dataset, designing the model architecture, and training and updating the model on the prepared data to reach a successful accuracy. The first step focused on the GTZAN dataset, a commonly used benchmark for music genre classification research.

2.1 GTZAN Dataset

This study used the GTZAN dataset as a basis for the training and testing of the model. This dataset is a collection of songs that fall into 10 different genres including blues, classical, country, disco, hip hop, jazz, metal, pop, reggae, and rock. Each song has both a 3-second and 30-second audio file, as well as a Mel spectrogram, representing the frequency of the sound clip over time [3]. This allowed both audio of the song and visual representation. An example of a Mel spectrogram that is depicting a blues song is shown in Figure 1 below.

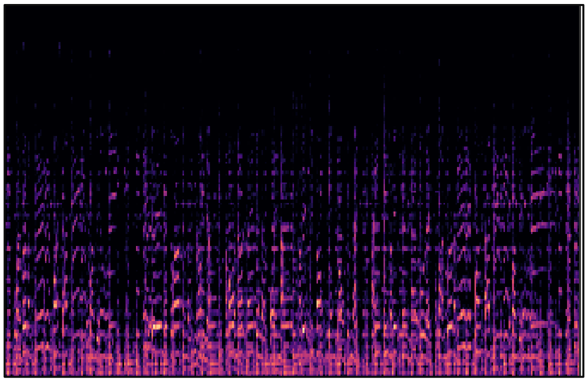


Figure 1: Mel Spectrogram – blues00000.png [1]

The dataset also offered a CSV file including information extracted from the 30-second audio files for each song. This data included some information that could be derived from each spectrogram as numerical form, like spectral centroid or bandwidth, that determine the energy of the music. It also included information found in the raw audio file. For example, the "chroma" features look at the actual notes or harmonies being used [5]. The CSV file also contained the labels of each audio file, which was necessary for testing and training the model [6].

2.2 Preprocessing the Data

To work with the data effectively, all of the images were made into 128 x 128 size, and three channels were used. This allowed for the full use of the spectrogram, as the color indicated levels of frequency and emphasized certain patterns. In Figure 2 below, the higher frequencies are shown in yellow, whereas the lower frequencies are darker colors, like purple.

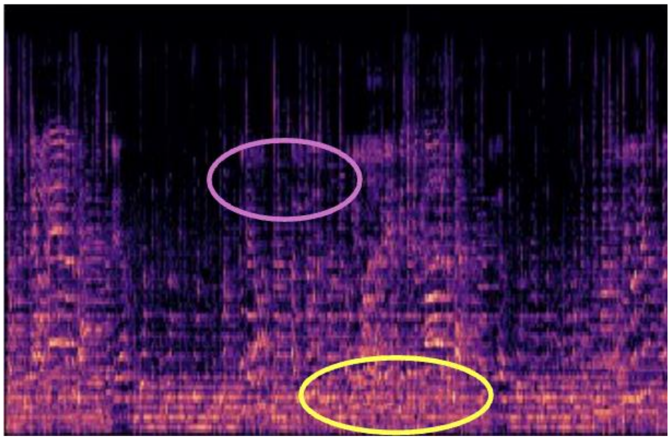


Figure 2: Mel Spectrogram Frequency Distributions

It was important that the weights of the frequencies were normalized because the higher frequencies were overpowering the lower frequencies in the data. Patterns can be found in both high and low frequencies, so this uneven distribution created issues in the accuracy of the models, as it would ignore the lower frequency patterns and put an extreme emphasis on the high frequencies.

2.3 Model Architecture

Based on the data the GTZAN dataset provided, there were two main approaches when handling the data and classifying the music: (1) parsing through the spectrograms to find important patterns or features, and (2) using the provided CSV file and predetermined data, like the mean, variance, and bandwidth of each clip to support the pattern findings from the spectrogram and bolster the results to increase accuracy.

Looking at images and finding patterns to make a prediction is done through classification, and a strong method of accomplishing this is by using a Convolutional Neural Network (CNN). However, CNNs would not be able to accommodate the addition of the audio file features. To mitigate this, another deep learning algorithm, Multi-Layer Perceptron (MLP) was brought in to provide support and enhance the learning of the model.

2.3.1 CNN Architecture

A CNN is a strong model used to process images through a system of different layers and filters to dig deeper into what is defining the image used as input. These layers allow the model to determine patterns and extract features that can be used to identify what the image is depicting. This inherently allows a prediction to be made based on the features. This was a great candidate for this experiment because the CNN model is able to parse through each spectrogram and find patterns in the music. It is able to see which genres the music aligned with the most, considering how many patterns were replicated in each genre in comparison to one of the spectrograms.

The model used in this experiment was built using a combination of convolutional layers, using the ReLU activation function, and pooling layers. Each convolutional layer was built using a 3x3 filter. The small filter size allowed the network to capture fine-grained local features within the spectrograms, which work well when followed by a maxpooling layer which reduced the dimensionality of the data, while keeping the most important, prominent features.

2.3.2 Multi-Layer Perceptron (MLP) Architecture

An MLP is a feedforward neural network that is built from fully connected, dense layers. These layers all work together to learn and create different, complex combinations of the input variables to make predictions. This was implemented in order to handle the features from the audio data file. Rather than looking at the spectrogram images, the MLP was implemented to try different combinations of the features and see which ones led to accurate predictions.

The MLP was implemented using three dense layers, followed by a dropout layer. The dropout layer was essential for reducing overfitting by randomly deactivating a portion of the neurons during training. This prevented the model from relying too heavily on any one combination of features and encouraged it to generalize better to unseen data.

Overall, adding this part of the model ensured that the songs were looked at from different angles, rather than only spectrograms.

2.3.3 Combined Model

A representation of a final model is shown below in Figure 3. The top section of the diagram depicts the CNN model, where the spectrograms are inputted, and the bottom section shows the audio feature file being inputted into the MLP. Once each individual section is complete, the results are combined into a dense layer, that again, will make combinations to find the best combinations of the individual models. This is followed by a dropout layer that reduces the number of combinations and flows into the final dense layer that will make the prediction of the genre label.

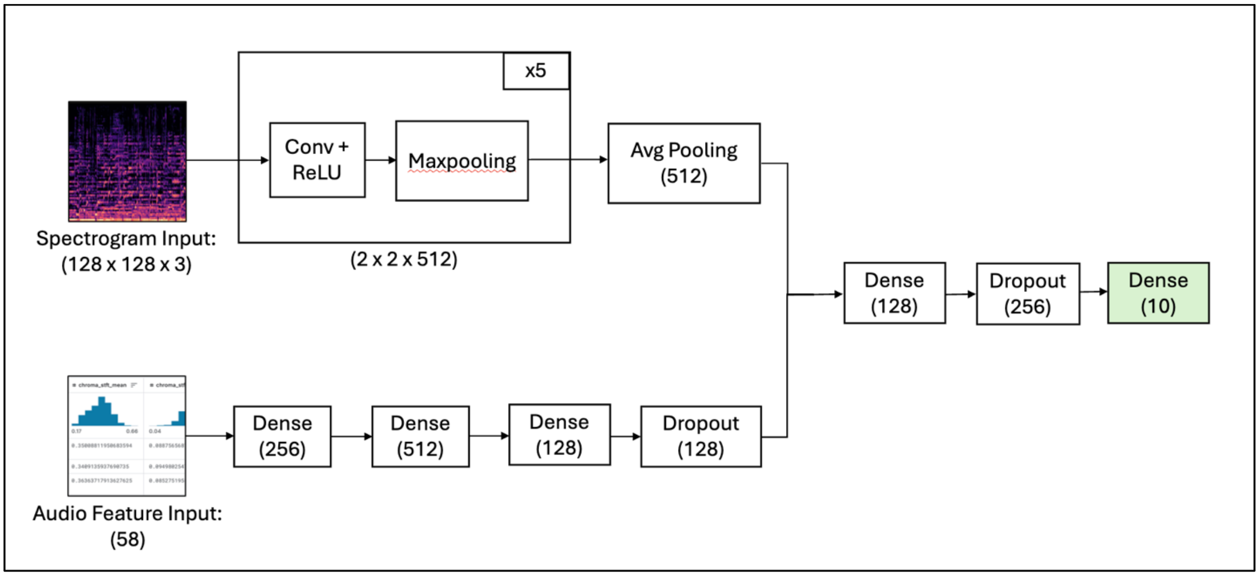


Figure 3: CNN and MLP Combined - Classification Algorithm

2.4 Training and Testing the Model on the Dataset

The dataset was split into training and validation sets. Here, 80% of the dataset contributed to training the model, whereas 20% was used as validation. The data was split using “random_state” = 42, which would ensure that each time the model was trained it would use the same split of the data. This allowed the results to be repeatable and ensured that new data wasn’t being trained every time the model was trained. It was also ensured that there would be a balanced distribution of each genre being examined by using “stratify=y_encoded”, where y_encoded was the numerical representation of each of the genres (blues = [0], rock = [9], etc.) These steps allowed for clean data to be used in model and helped with accuracy of the results.

In addition, the model was trained with a batch size of 32 over 20 epochs. The model was built using Adam as the optimizer and categorical cross-entropy loss. These worked together to update the model during training, where the loss would indicate to the Adam model what was causing the most loss and it would change weights based on that input [6].

3 Related Work

Music genre classification is a popular research topic and has been widely studied. Researchers have studied many different approaches, including using different datasets with different types of spectrograms, a variety of deep learning architectures, and transformed the output to go beyond just classifying the music to a certain genre. Learning from some of these studies gave good insight into what truly can be done with music classification through deep learning models.

3.1 End-to-End Learning for Music Audio Tagging at Scale by Pandora

Researchers at Pandora, another music streaming service, developed a deep learning system that used a deep CNN model to create tags for a large dataset of songs. Pandora's business model, like Spotify, offers personalized music suggestions that rely heavily on classification of songs. However, rather than stopping at just genre classification, this algorithm tags, or adds learned labels, to songs based on a variety of identifiers. These could be in the form of the instruments being played, the mood of the music, and many more. Being a streaming service, Pandora has access to incredibly large datasets, comprised of over 1.2 million songs. This allowed training over 1200 times the number of songs than were in the GTZAN database. It is confirmed in the article that the best results came from the model that was trained with over 1 million songs and took slightly less than two weeks to fully train. Overall, this model achieved very impressive results, with a ROC AUC of about 89%, which means that model distinguishes between classes very effectively. While our research was inspired by this work, it was conducted on a much smaller scale due to the limitations in the size of the dataset and a fixed set of labels to work with [7].

3.2 Automatic Tagging Using Deep Convolutional Neural Networks

A similar study was done out of Queen Mary University of London. Like the model done by Pandora, this study looked at tagging the audio files with multiple attributes. The study focused heavily on model architecture, detailing design choices such as the size of convolutional filters, the use of pooling layers, and data normalization techniques. Their approach showed that using Mel spectrograms with a carefully designed CNN could achieve a very high-level performance, especially when trained on large-scale datasets [8].

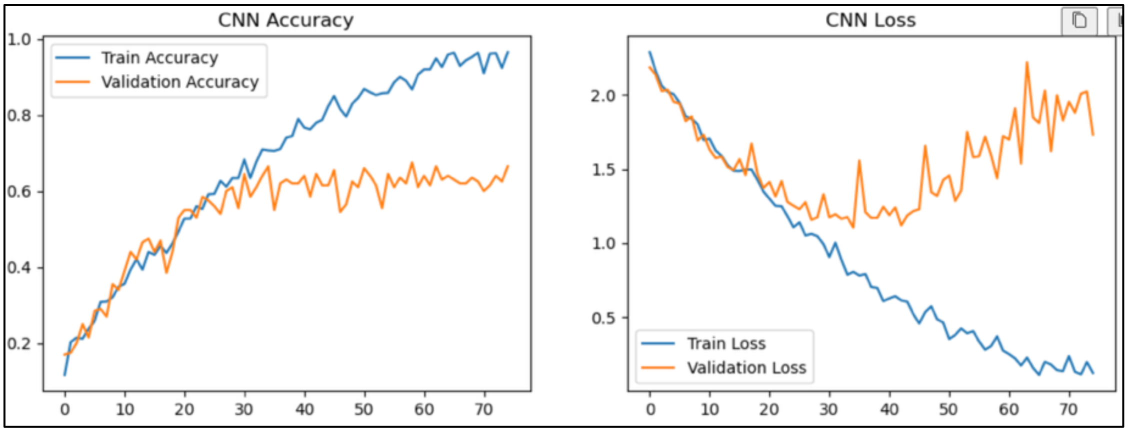
The model used in this study was structurally similar to ours in that it relied on convolutional layers to extract patterns from Mel spectrograms; however, their architecture focused solely on spectrogram input for multi-label tagging, whereas our implementation combined spectrograms with other audio features for single-label genre classification. Even with some of the differences between them, the model depicted in this study shows that the model we implemented is an effective approach for learning meaningful representations from audio data.

4 Experimental Results

The project initially focused on using a CNN to classify music genres based solely on visual patterns found in Mel spectrograms. However, during testing, the model's limited performance suggested that incorporating structured audio features could improve accuracy. This led to the development of a hybrid model that combined the strengths of both CNN and MLP architectures.

4.1 Convolutional Neural Network (CNN) Model

The first model attempted was the CNN model by itself. The model was made up of 14 layers that incorporated down sampling, increasing filter size through each block, and eventually flattening the data as we had done/seen throughout different assignments. This allowed us to make a simple CNN that could take in the most information from the images of the spectrograms. When testing this model, we discovered that while the training accuracy was decent at around 0.6, it was clear that after about 30 epochs the validation accuracy stagnated while the loss exploded as seen in Figure 4(a) and 4(b). This behavior indicated that the data was likely overfitting, which was most likely caused by a lack of variability in the dataset. It was also lacking the additional context that the audio features provided.



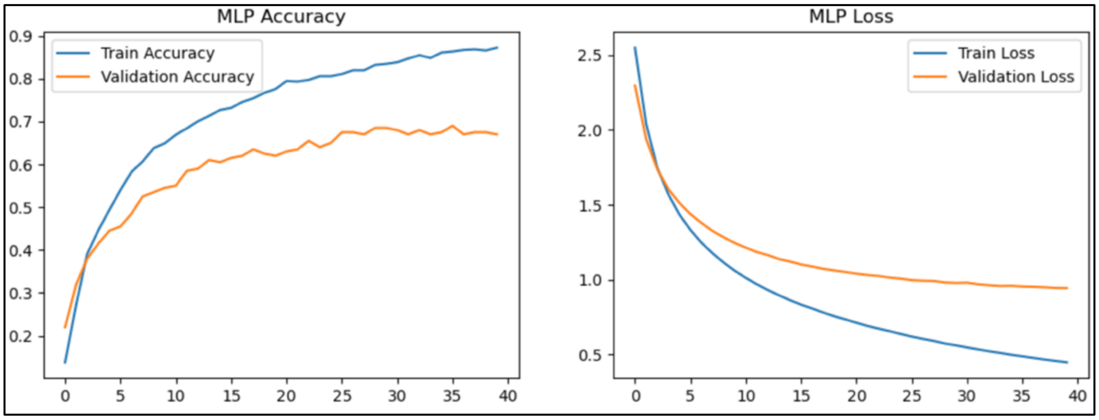
(a) **(b)**

Figure 4: Exclusive CNN Training and Validation (a) Accuracy and (b) Loss

While the CNN did not give optimal results, it gave a launching point for the rest of the hybrid model. The same is true for the MLP individual model which is outlined below.

4.2 Multi-Layer Perceptron (MLP) Model

The next thing that we wanted to test was the accuracy and loss of a MLP on its own. Shown in Figure 5, the validation accuracy quickly reached 0.5 but then stagnated around 0.65. The loss quickly fell to 1.25 and stagnated, showing that the MLP wasn't overfitting like the CNN had. These results suggested that while the MLP alone was also limited in accuracy, it did not overfit like the CNN model and would combine well with the CNN in a hybrid model.

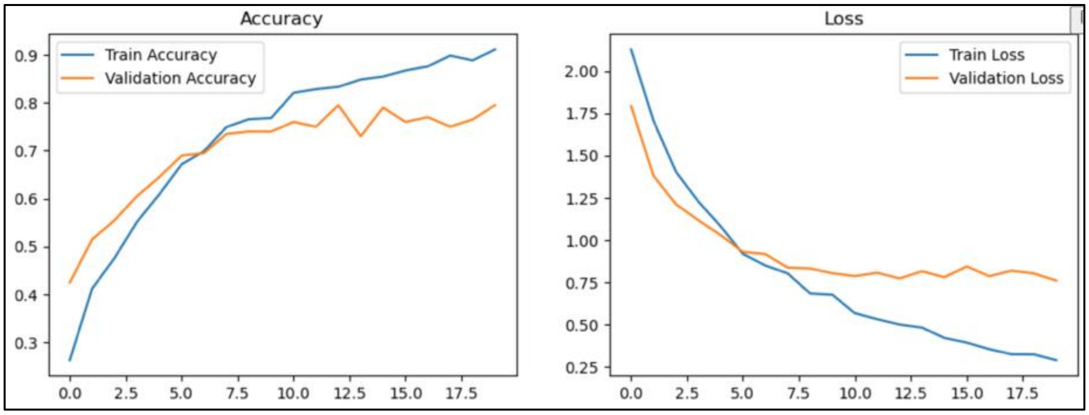


(a) **(b)**

Figure 5: MLP Training and Validation (a) Accuracy and (b) Loss

4.3 Final Hybrid Model

The hybrid model was trained over 20 epochs and the results surpassed each of the individual models with less training. As shown in Figure 6(a) and 6(b) respectively, both the validation accuracy and loss showed significantly better results as compared to either individual model with the loss staying flat showing that overfitting was not an issue.



(a) **(b)**

Figure 6: Hybrid Model Training and Validation (a) Accuracy and (b) Loss

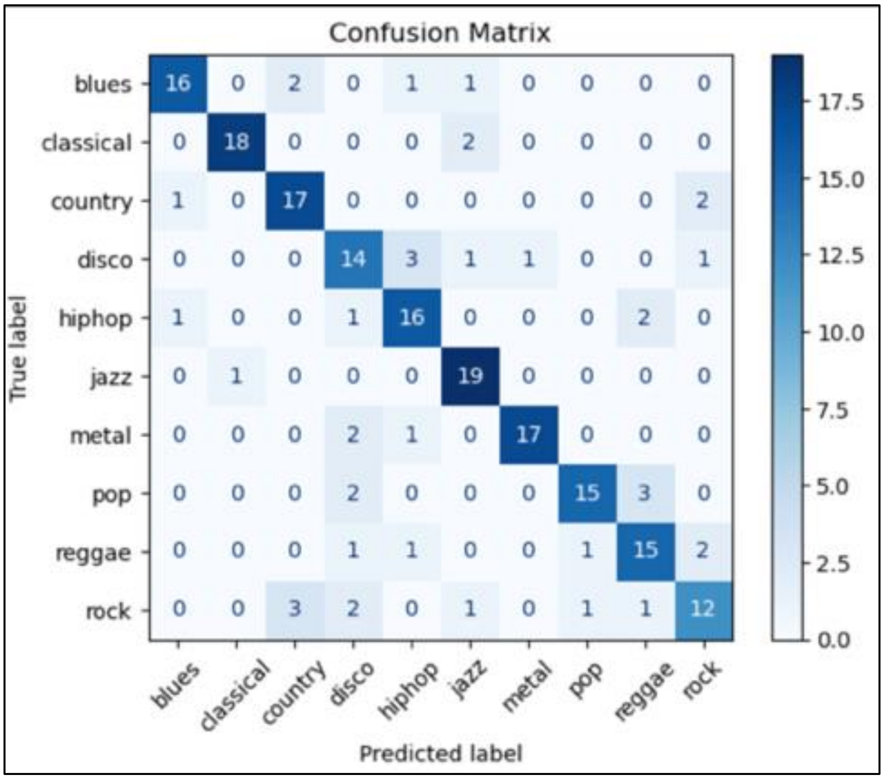
To further analyze the performance of the hybrid model we added functionality to see the statistics of different accuracy measurements in regard to individual genres as well as a weighted total. In Table 1 rock and disco had the worst F1-score while classical and metal had the best F1-score. This is most likely due to more distinct patterns. For example, metal music is usually high-energy with bigger spikes in frequency that are not commonly replicated, which is similar to classical which can have a wide frequency range.

Table 1: Accuracy Statistics for Individual Genres

	Precision	Recall	F1-Score	Support
Blues	0.89	0.80	0.84	20
Classical	0.95	0.90	0.92	20
Country	0.77	0.85	0.81	20
Disco	0.64	0.70	0.67	20
Hip-hop	0.73	0.80	0.76	20
Jazz	0.79	0.95	0.86	20
Metal	0.94	0.85	0.86	20
Pop	0.88	0.75	0.86	20
Reggae	0.71	0.75	0.73	20
Rock	0.71	0.60	0.65	20

Additionally, the F1-accuracy for the entire dataset ended up being 0.8. Figure 7 indicates that the most mistaken genre was rock, and it was most often misclassified as country. This explains the rock genre's F1-score being the lowest and correlates to real life since country and rock can have often overlapping patterns and sounds. Even with this overlap, the model largely guessed correctly over all genres.

This accuracy of the model is best shown through the confusion matrix in Figure 7 below where each prediction is shown visually. In this matrix, all of the darker blue squares indicated that prediction was correct and that there was a high number of correct predictions. These numbers ranged from 12 to 19, which significantly outweighed the incorrect predictions.



	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	16	0	2	0	1	1	0	0	0	0
classical	0	18	0	0	0	2	0	0	0	0
country	1	0	17	0	0	0	0	0	0	2
disco	0	0	0	14	3	1	1	0	0	1
hiphop	1	0	0	1	16	0	0	0	2	0
jazz	0	1	0	0	0	19	0	0	0	0
metal	0	0	0	2	1	0	17	0	0	0
pop	0	0	0	2	0	0	0	15	3	0
reggae	0	0	0	1	1	0	0	1	15	2
rock	0	0	3	2	0	1	0	1	1	12

Figure 7: Confusion Matrix of True Genre vs Predicted Genre

These results confirmed that combining both spectrogram-based image analysis and structured audio features led to a significantly more accurate and balanced model for music genre classification.

5 Conclusion

This project allowed for exploration of music genre classification in multiple different ways. Using the GTZAN dataset, three models were implemented and evaluated: a Convolutional Neural Network (CNN), a Multi-Layer Perceptron (MLP), and a hybrid model that combined both approaches. The CNN implementation by itself demonstrated issues of overfitting and high variance in its accuracy, while the MLP was more consistent but failed to reach the anticipated performance accuracy. The final hybrid model, however, was able to integrate both of the other models to create an accurate and consistent performance, labelling genres with a F1-score accuracy of 0.80. Additionally, we were able to properly show the statistics of every genre's prediction accuracy and discern what genres were most mistaken for others.

While we were successful in terms of reaching the goal set at the beginning of the semester, of an accuracy of at least 70%, there are still more opportunities for improving the model's performance. Some possible solutions could be changing the structure of the CNN to utilize batch normalization, add learning rate scheduling, and possibly trying gradient boosting. We would not try using data augmentation though since this will likely result in more misclassification between already closely correlated genres. Although there is still much to explore, we believe this project made significant progress. It offers a strong foundation for further experimentation and improvement in future work.

Contributions

Task	People
Project Proposal	Emily, Matt, Ebube
Prepressing Data	Emily, Matt, Ebube
Implementing Algorithm #1 - CNN Only	Emily, Ebube
Midpoint Report	Emily, Matt, Ebube
Implementing Algorithm #2 – CNN/MLP	Ebube, Matt
Evaluating Al Models and Defining Results	Emily, Matt, Ebube
Slides, Demo, and Presentation	Emily, Matt, Ebube
Writing Report	Emily, Matt

Acknowledgements

We would like to acknowledge the open-source libraries and tools that supported our work on this project, as well as the creators of the GTZAN dataset, which formed the basis of our analysis.

References

[1] Spotify for Developers. (n.d.). *Get audio features for a track*. Spotify. https://developer.spotify.com/documentation/web-api/reference/get-audio-features

[2] The Data Scientist. (n.d.). *Understanding audio analysis: How Spotify classifies millions of tracks using AI*. https://thedatascientist.com/understanding-audio-analysis-how-spotify-classifies-millions-of-tracks-using-ai/

[3] Splice. (n.d.). *What is a spectrogram?* https://splice.com/blog/what-is-a-spectrogram/#:~:text=They%20may%20look%20intimidating%20at,high-frequency%20chirps%20and%20clicks

[4] Oluyale, D. (2019). *Detecting musical key from audio using chroma feature in Python*. [3] Medium. https://medium.com/@oluyaled/detecting-musical-key-from-audio-using-chroma-feature-in-python-72850c0ae4b1

[5] Olteanu, A. (n.d.). *GTZAN Dataset: Music Genre Classification*. Kaggle. https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

[6] Keras. (n.d.). *Adam optimizer*. https://keras.io/api/optimizers/adam/

[7] Choi, K., Fazekas, G., & Sandler, M. (2016). *Automatic tagging using deep convolutional neural networks*. In Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR). https://arxiv.org/pdf/1606.00298

[8] Pons, J., Nieto, O., Prockup, M., Schmidt, E., Ehmann, A., & Serra, X. (2017). *End-to-end learning for music audio tagging at scale*. In Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR). https://arxiv.org/pdf/1711.02520

Link to Code: https://drive.google.com/drive/folders/1AMhAfs6ElKQTX0cCuqirlsNv1reKwaf2?usp=drive_link