

Final Project Report

Music Genre Classification


Group 43

Surya Pratap Singh
Shaurin Karnik

singh.sury@northeastern.edu
karnik.sha@northeastern.edu

Percentage of Effort Contributed by Student 1: 100%

Percentage of Effort Contributed by Student 2: 100%

Signature of Student 1: 

Signature of Student 2: 

Submission Date: 04/21/2023

Table of Contents

S. No	Topic	Page Number
1.	Project Problem	3
1.a	Problem Setting	3
1.b	Problem Definition	3
2.	The Data	3
2.a	Data Source	3
2.b	Data Description	3
3.	Data Exploration and Visualization	4
4.	Exploration of Data Mining Models	9
5.	Model Performance Evaluation	13
6.	Final Decision for classification	21

1. Project Problem

1.a Problem Setting

In comparison to specialized archives and private sound collections, music databases are continuously improving their reputation. The number of users accessing the music database has increased along with advancements in internet services and increased network bandwidth. Extremely huge music libraries require a lot of time and effort to manage. Most of the music files are organized by artist or song title. This could make it difficult to find music that fits a particular genre. Blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock are among the most popular musical genres. Additionally, the concept of a musical genre has evolved over time. Pop songs created fifty years ago, for example, differ from pop songs created today. Machine Learning techniques are utilized to accomplish Automatic Music Genre Classification because it is laborious to manually categorize each track.

1.b Problem Definition

The main objective is to run several machine learning models that categorize music samples into various genres using spectrograms and histograms in a more methodical manner, and discover which models are most accurate. Making the selection of songs easier and quicker is the goal of automating the music classification. If one must manually categorize the songs or music, they must first be listened to a great deal before the genre is chosen. This takes a lot of time and is challenging. Automating music classification can make it easier to locate important information like trends, popular genres, and performers. The first step in this direction is identifying music genres.

2. The Data

2.a Data Source

The dataset considered for this project is taken from Kaggle. Kaggle is an open-source platform for datasets and other related fields in Data Science. Refer to the [link](#) to dataset for this project.

2.b Data Description

The dataset has total 9990 rows and 60 columns which focuses on different attributes and parameters for songs of different genre types. All the columns in the dataset have their key significance. The different genres that the dataset focuses on are blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. The key parameters required for the visualizations from this dataset include Chroma STFT Mean, RMS Mean, Spectral Centroid Mean, Spectral Bandwidth Mean, and MFCC (Mel Freq. Cepstral Coefficients), and the variance of the MFCC for several ranges. These factors capture different aspects of the audio signal, such as pitch, loudness, and timbre, which are important features for characterizing music genre.

3. Data Exploration and Visualization

It's also important to consider the labels or genres themselves, as they will be the target variable for the visualization. The label column in the dataset should be used to group or color the data points according to their genre, so that we can see how each factor varies by genre. The project is based on classification of music genres of several different categories like Jazz, Classical, Disco, etc. The dataset has a total of 9990 rows and 60 columns, which has all the essential features of a audio.

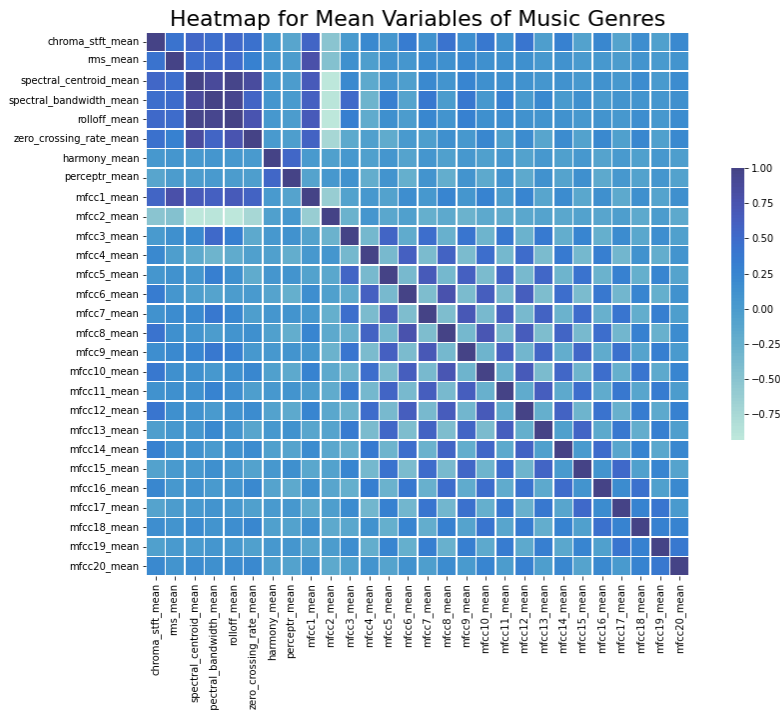


Fig 1: Heatmap for mean variables of Music Genres

The above plot is a heatmap visualization which determines the correlation for all the essential attributes/ parameters of audio files based on the genres. The intensity of the color pattern represents the correlation range for all attributes. From the above heatmap, there exists variables that are highly correlated, but because all the columns make a significant difference for the classification of music genre classification, thus these columns cannot be dropped.

The graph below (figure 2) is a line graph representation of the harmonic range of the of song belonging to a particular label category. By the above line graph, it can be inferred that the range of the harmonious factor are different for all the label categories. Pop is the type of label category that has the highest harmonic range, whereas on the other hand, reggae label is the category that has the lowest harmonic range.

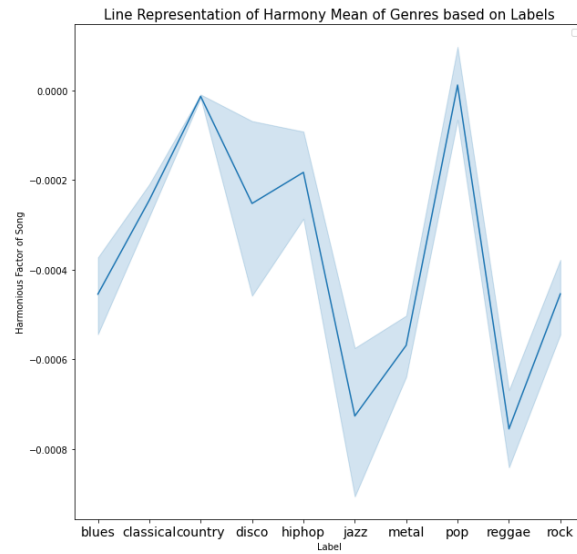


Fig 2: Line representation of harmony mean of genres based on labels



Fig 3: Scatter Plot for Chroma vs RMS

The above graph is an interactive scatter plot that allows to understand the insights of Chroma vs RMS value of the different label categories. Although most of the label categories have similar type of range, there exists an exception as outliers in labels reggae and classical. The graph would have been better to understand if it would have been an interactive one.

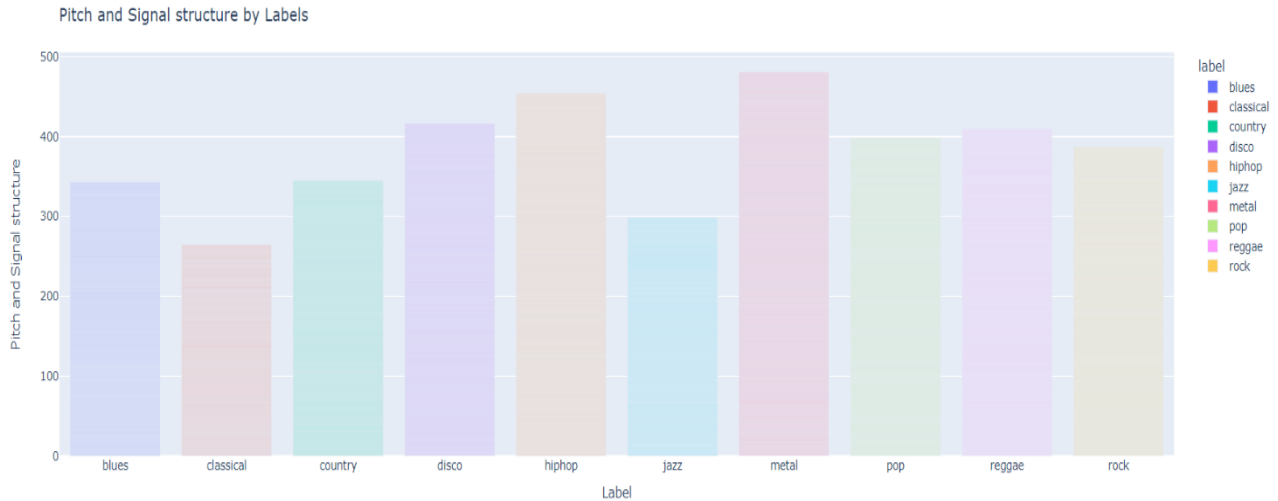


Fig 4: Bar graph representation for pitch and signal structure by labels

The above visualization is a bar graph that shows the pitch and signal structure of each label categories respectively. By the bar graph, it can be observed that the label category- metal has the highest pitch and signal structure, whereas classical category has the lowest

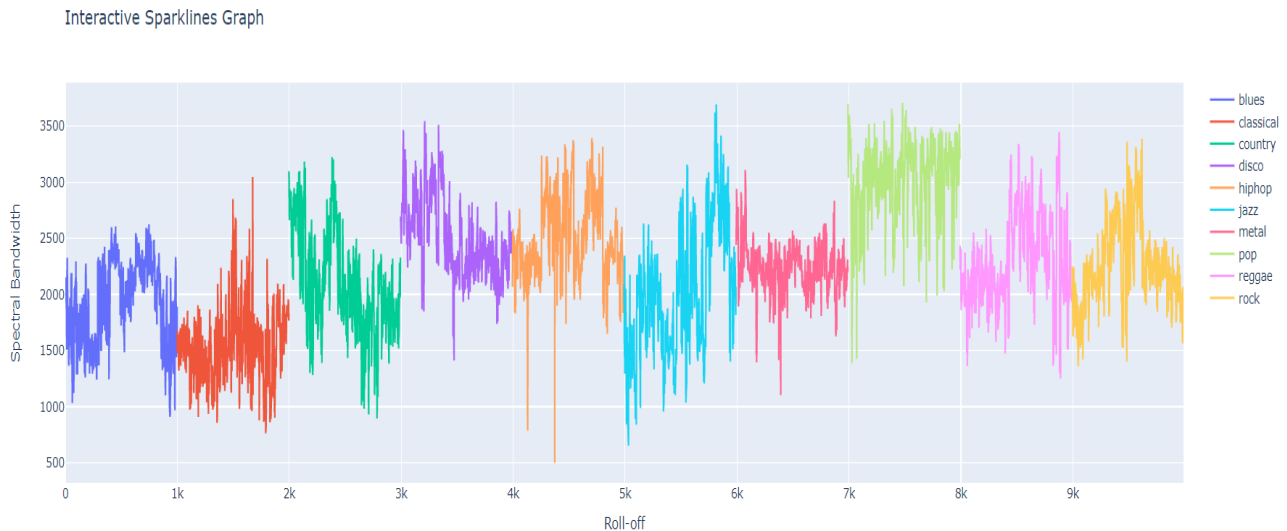


Fig 5: Sparkline representation for music genres

The above graph is a sparkline graph representation of different genres categories based on Spectral Bandwidth and Roll-off parameters. By above graph, it can be concluded that the distribution in case of categories like disco, hip hop, jazz, pop, and rock have a continuous huge increasing and decreasing pattern. It is obvious to have such representation for these genre cases since these type of genre categories are the categories that drops the beat at sudden and then the next moment, there is a new beat with greater intensity.

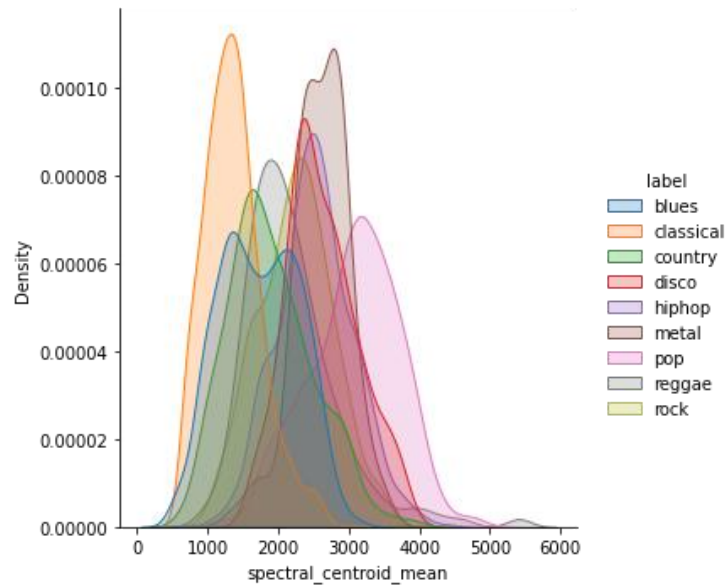


Fig 6: Density graph for Music Genres based on Density and Centroid Mean

The above graph is a distribution graph based on Density of the song and the spectral centroid mean values for every genre category. From the graph, it can be concluded that classical and metal has almost the similar type of the distribution. From the above graph, it can be also inferred that the density of the genre type typically depends on the spectral centroid.

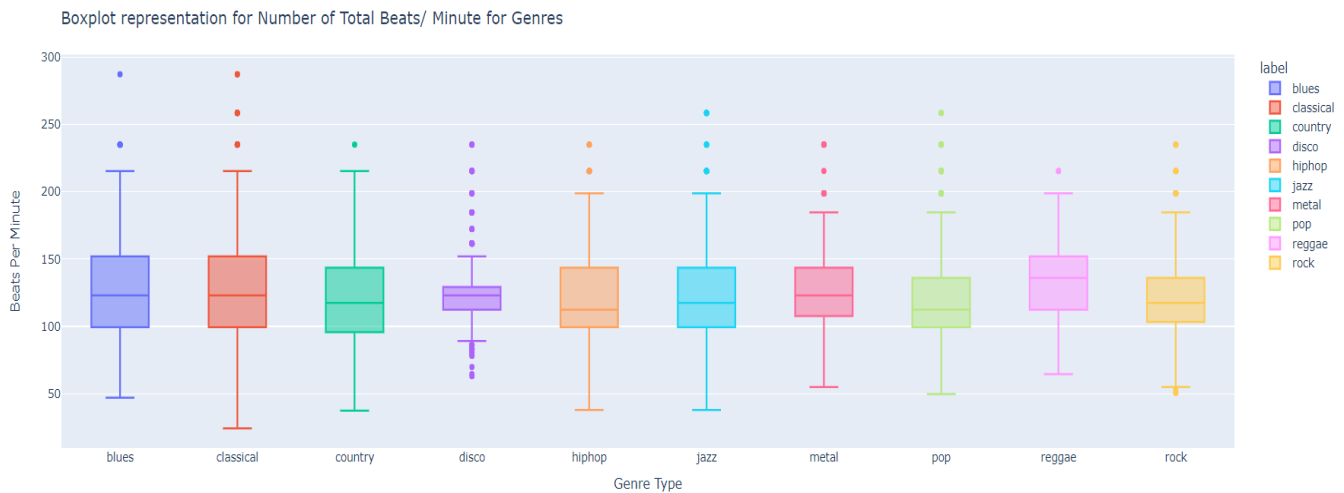


Fig 7: Boxplot representation for total beats per minutes

The above graph is a box plot representation of total beats per minute based on different genre categories. The number of beats depends on the type of genre. From the graph above, it can be inferred that classical has the highest number of the beats per minute and that is obvious to have because classical and blues genres are the types of genres that have the beats for almost entire songs.

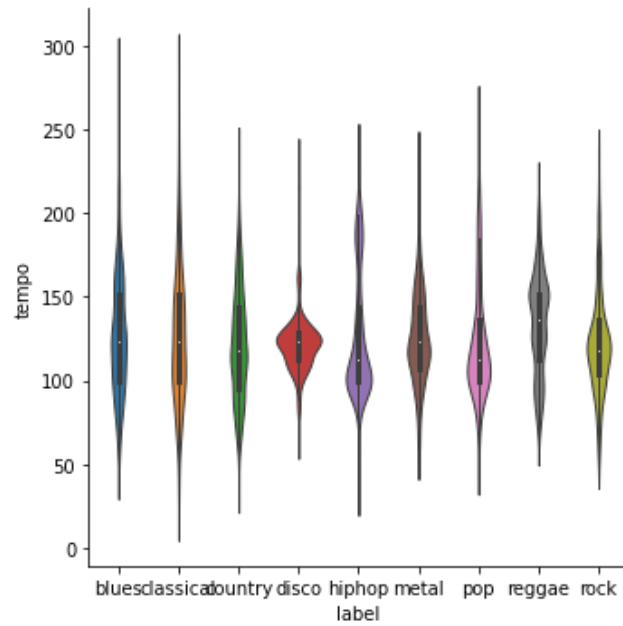


Fig 8: Violin representation for total beats per minutes

The above graph is a violin graph representing tempo of the song based on label categories for each. From the above violin graph, categories like blues, classical, and country have almost the same structure. However, labels like disco, hip-hop, pop, and rock are labels that have different structure from the above-mentioned labels.

Note: To understand violin plot, it is important to have box plot representation based on the same parameters. Thus, the box plot and the violin plot are visualized using the same parameters, that is tempo and labels.

Thus, by the above graphical visualizations, it can now be clearly understood that each label category (genre) has a different understanding and representation. Although the labels might look similar by the way they are visualized, each label (genre) has its own significance with each type having its own attribute values that differ from one another.

Dimension reduction techniques are not applicable for this dataset since this dataset is completely focused on parameters of an audio files and the exact audio files. Thus, the main focus of this project is for the classification of music genre system using neural networks. The audio files will be converted to multiple copies in chunks, and based on those chunks, the histogram and spectrogram plots will be created for those chunk files of each audio files. Then by using the histogram and spectrogram, we will classify the music genres.

4. Exploration of Candidate Data Mining Models and select the final Model or Models

In the context of this project, the image presented above visually displays the various categories of music genres that will be utilized. These genre categories serve as a fundamental aspect of the project's completion, with the intention of providing a clear and concise framework for analyzing and categorizing music based on its genre. The image serves to illustrate the specific music genre categories that will be employed, thereby serving as an informative and helpful reference for those engaged in the project's completion.

The dataset under consideration comprises a total of 900 audio files, encompassing nine distinct music genres, with each genre containing 100 audio files. To facilitate further analysis, each audio file within every genre category will be converted into ten separate chunks, resulting in a total of 9,000 individual audio files. This conversion process entails dividing each audio file into ten segments, each of which constitutes a separate and distinct chunk. By undertaking this step, the dataset will become better suited for machine learning algorithms, as it provides a more granular and detailed representation of the audio files.

The python code snippet facilitates the simultaneous selection of all music genres, subsequently converting each corresponding audio file value into ten discrete, distinct chunks of the same audio file. This implementation allows for the efficient and streamlined processing of all genres, reducing the potential for variability in results and improving the overall accuracy of the analysis.

With the completion of the audio file chunking process, the next step involves generating both histograms and spectrograms based on the newly generated audio files. These visualizations serve as critical components in the classification of music genre types, achieved through the creation of a self-designed neural network. This model represents the preferred approach for project completion, given its capacity to analyze the vast and complex dataset generated.

While other deep learning models like **VGG**, **MobileNet**, **LeNet**, and others could be potentially compared together with self-designed Neural Network, the extensive time requirements for execution (**generally between 12 to 24 hours non-stop**) renders this approach impractical. Thus, the primary focus remains on designing a self-neural network capable of processing the large audio dataset effectively and comparing it with **MobileNet** model, albeit with a considerable amount of time for computation. Given that Neural Networks also take a higher time for the execution.

Preferred Final Model for the Project: Neural Networks (Self CNN & MobileNet)

Neural networks are a class of machine learning algorithms inspired by the structure and function of biological neural networks in the human brain. They are used for a variety of tasks such as image recognition, natural language processing, and speech recognition. A neural network is also considered as a neuron system and consists of interconnected nodes or neurons organized into layers. The input layer receives the input data, which is then processed by one or more hidden layers, and finally, the output layer produces the result. Each neuron takes in one or more inputs, multiplies them by a weight, adds a bias term, and applies an activation function to produce an output. The

weights and biases are adjusted during training to optimize the network's performance on a given task.

There are several types of neural networks, including feedforward neural networks, convolutional neural networks, recurrent neural networks, and generative adversarial networks. Each type of neural network is designed for a specific task and has its own architecture and training techniques.

The training process for a neural network involves feeding it a large amount of labeled data and adjusting the weights and biases to minimize the difference between the network's output and the true labels. This is typically done using an optimization algorithm such as gradient descent. Neural networks have proven to be very powerful and versatile tools for a wide range of machine learning tasks. However, they can also be very complex and computationally intensive, requiring large amounts of training data and computational resources to achieve good performance.

Advantages/ Pros of Neural Networks

Few of the advantages/ Pros of Neural Networks are:

1. **Flexibility:** Neural networks are very flexible and can be used for a wide range of tasks, including classification, regression, and prediction.
2. **Non-linearity:** Neural networks can model complex, non-linear relationships between inputs and outputs. This makes them very useful for tasks such as image recognition and natural language processing.
3. **Adaptability:** Neural networks can adapt to new data and learn from experience. This means that they can continue to improve their performance over time as more data becomes available.
4. **Fault tolerance:** Neural networks are able to continue functioning even if some of their components fail. This is because they are designed to be robust and able to tolerate noise and errors in the input data.
5. **Scalability:** Neural networks can be scaled up to handle very large datasets and complex models. This makes them very useful for tasks such as speech recognition and natural language processing, which require a large amount of computational power.
6. **Automation:** Neural networks can automate many tasks that would otherwise require human intervention, such as image recognition and speech recognition. This can save time and resources and improve accuracy.
7. **Parallel processing:** Neural networks can be designed to run on parallel hardware such as GPUs, which can greatly speed up the training and inference process. This makes them very useful for applications that require real-time processing, such as autonomous driving and robotics.

Overall, neural networks are very powerful and versatile tools that can be used for a wide range of machine learning tasks. While they can be complex and computationally intensive, their flexibility and adaptability make them a valuable tool for many applications.

Disadvantages/ Cons of Neural Networks:

Some disadvantages or challenges of Neural Networks:

1. **Requires large amounts of data:** Neural networks require a large amount of data to train effectively. This can be a challenge for applications where data is scarce or difficult to obtain.
2. **Overfitting:** Neural networks can sometimes be overfit to the training data, which means that they perform well on the training data but poorly on new, unseen data. This can be addressed by regularization techniques, but it remains an ongoing challenge.
3. **Computationally intensive:** Neural networks can be very computationally intensive, especially for large datasets and complex models. This can require specialized hardware and can make training and inference very slow.
4. **Difficult to interpret:** Neural networks can be difficult to interpret, especially for very deep models with many layers. This can make it difficult to understand how the network is making its predictions, which can be a problem for applications where interpretability is important.
5. **Requires expertise:** Building and training a neural network requires specialized knowledge and expertise. This can be a barrier to entry for some users, especially those who are new to machine learning.
6. **Black box nature:** Neural networks can sometimes be seen as a "black box" because it is not always clear how the network is making its predictions. This can be a problem for applications where transparency and accountability are important.
7. **Vulnerable to adversarial attacks:** Neural networks can be vulnerable to adversarial attacks, where small changes to the input can cause the network to make incorrect predictions. This can be a problem for applications such as autonomous driving and cybersecurity.

Overall, while neural networks are very powerful and versatile tools for machine learning, they also come with a number of challenges and potential disadvantages. It's important to carefully consider these factors when deciding whether to use a neural network for a particular application.

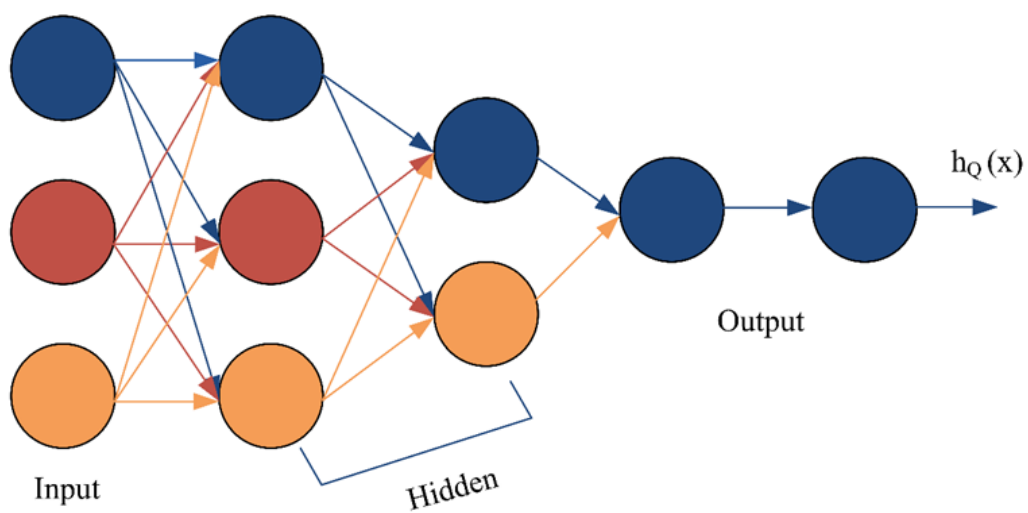


Fig 9: General Architecture of Neural Network

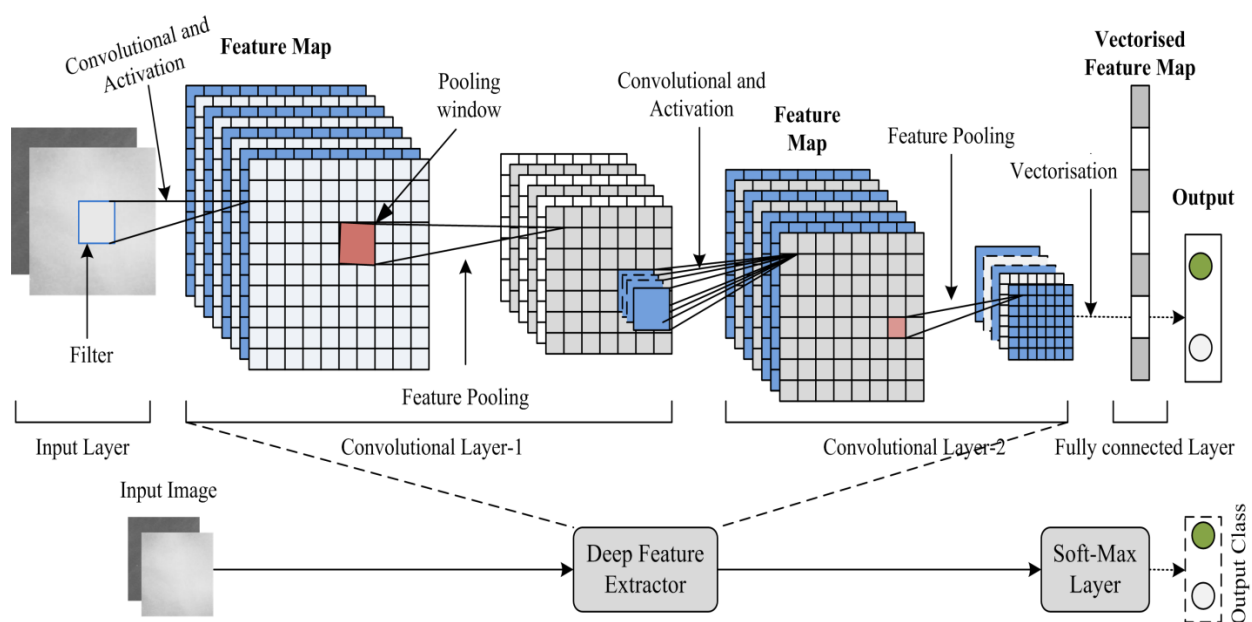


Fig 10: Detailed CNN Architecture

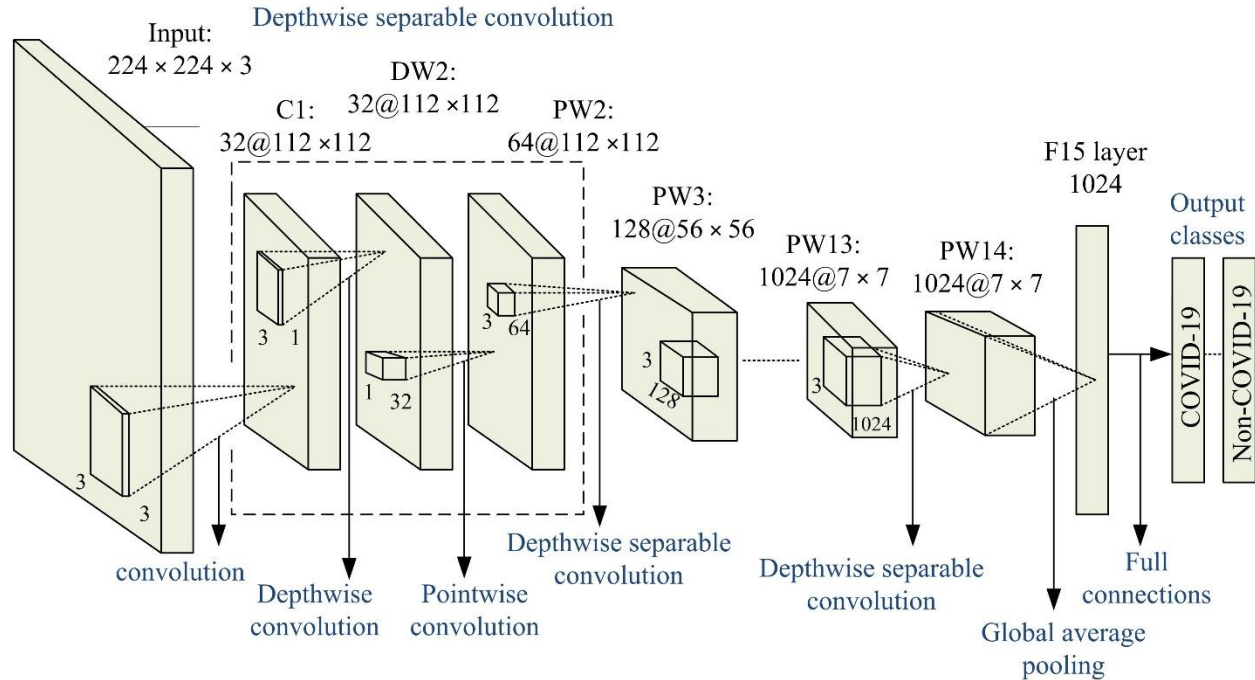


Fig 11: MobileNet Architecture

5. Model Performance Evaluation and Interpretation

In this project, we aimed to classify music genres based on different categories using two different models, namely self-designed neural network and MobileNet. Our dataset consisted of 10 different music genres, out of which one genre was removed due to its similarity to hip hop. Each genre had 100 audio files, resulting in a total of 900 audio files for all genres. We further segmented each audio file into 10 short audio files, generating a total of 9000 audio files for training and validation.

After processing the audio files, we generated separate histograms and spectrograms for all 9000 audio files. Thus, we obtained a total of 9000 audio files, 9000 histogram images, and 9000 spectrogram images. We split the samples into training and validation sets in a 75:25 ratio. The training set consisted of 6750 samples, and the validation set had 2250 samples.

We used the same amount of data for both the self-designed neural network and MobileNet models for training and testing. Our goal was to use these models to classify the different music genres based on the histogram and spectrogram images. Refer to table 1 for data bifurcation.

Table 1 Abstract details of the dataset bifurcation and its preparation.			
Considered class	No. of samples per class	Data bifurcation	
		Training	Validation
Blues	1000	750	250
Classical	1000	750	250
Country	1000	750	250
Disco	1000	750	250
Hip Hop	1000	750	250
Metal	1000	750	250
Pop	1000	750	250
Reggae	1000	750	250
Rock	1000	750	250
Total	9000	6750	2250

Let us focus on Spectrograms and Histograms of each category respectively.

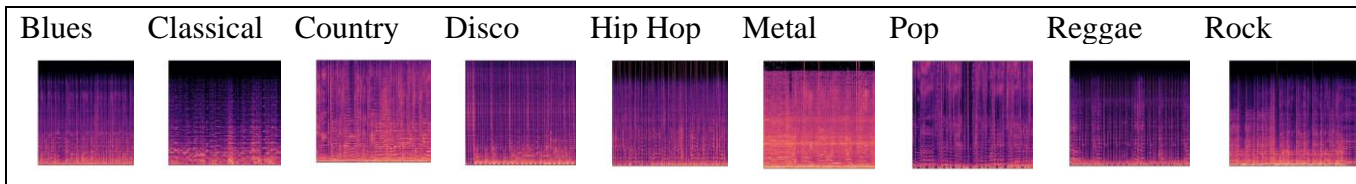


Fig 12: Spectrogram images samples for each category

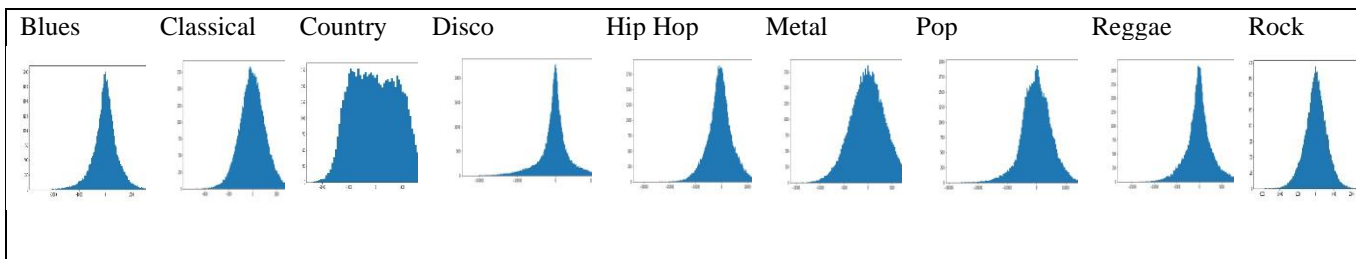


Fig 13: Histogram images samples for each category

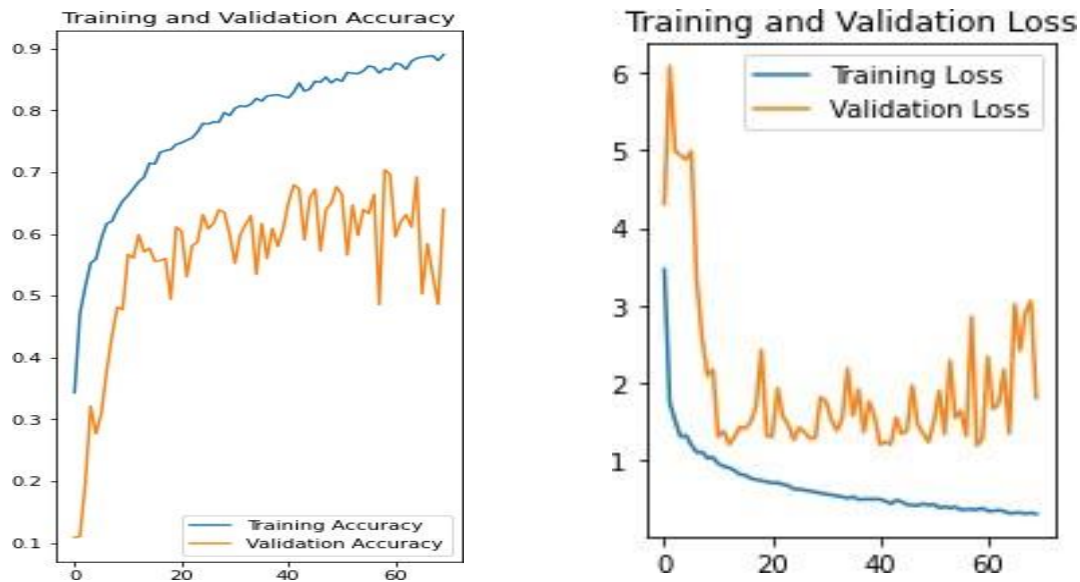


Fig 14: Accuracy and Loss for Spectrogram using Neural Network

After processing the audio files and converting them into spectrogram images, we fed these images into our custom-designed convolutional neural network (CNN). The neural network was trained on a dataset, and during the training process, we obtained a training accuracy of 89.63%, indicating that the model was able to correctly classify 89.63% of the data samples in the training set.

After training, we evaluated the performance of the model on a validation set, which was a subset of the original dataset that the model had not seen during training. The model achieved a validation accuracy of 65.44%, which is lower than the training accuracy, but still suggests that the model is able to generalize to unseen data.

We also calculated the accuracy loss and validation loss during the training process. The accuracy loss was found to be 0.7109, which indicates the amount of error between the predicted output of the model and the true output in the training set. Similarly, the validation loss was found to be 1.7130, which measures the error between the predicted output and the true output in the validation set. These values provide a measure of the overall performance of the model, and can be used to optimize the model further.

After processing the audio files and generating spectrogram images, we trained our MobileNet model on the dataset. The results revealed that the training accuracy was 74.82%, meaning that the model correctly classified 74.82% of the data samples in the training set.

To evaluate the model's performance on unseen data, we used a validation set, which the model had not encountered during training. The model achieved a validation accuracy of 59.43%, indicating that it was able to correctly classify 59.43% of the data samples in the validation set.

Furthermore, we calculated the accuracy loss and validation loss during the training process. The accuracy loss was found to be 0.7590, representing the error between the predicted output and true

output in the training set. The validation loss was 1.3480, which indicates the error between the predicted output and true output in the validation set.

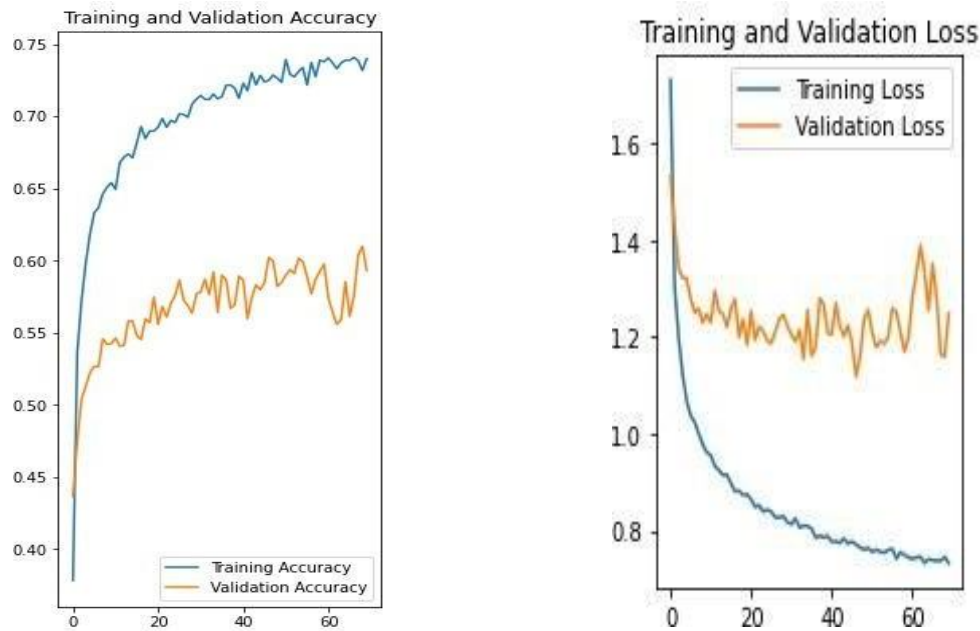


Fig 15: Accuracy and Loss for Spectrogram using MobileNet

Refer to table 2 for the tabular analysis of the comparison of both the models when trained for the spectrogram images.

Table 2

Achieved results of extensive experiments carried out for the work for Spectrograms of Audio Files

Model Name	Training Phase			Validation Phase		
	Number of cases	Accuracy (%)	Loss	Number of cases	Accuracy (%)	Loss
Neural Network	6750	89.63	0.7109	2250	65.44	1.7130
Mobile Net-V2	6750	74.82	0.7590	2250	59.43	1.3480

In the next section, we will now train the histogram images for all the cases on both the models and then evaluate the performance of both the models and then evaluate their comparison and then decide the winning model to predict the final classification.

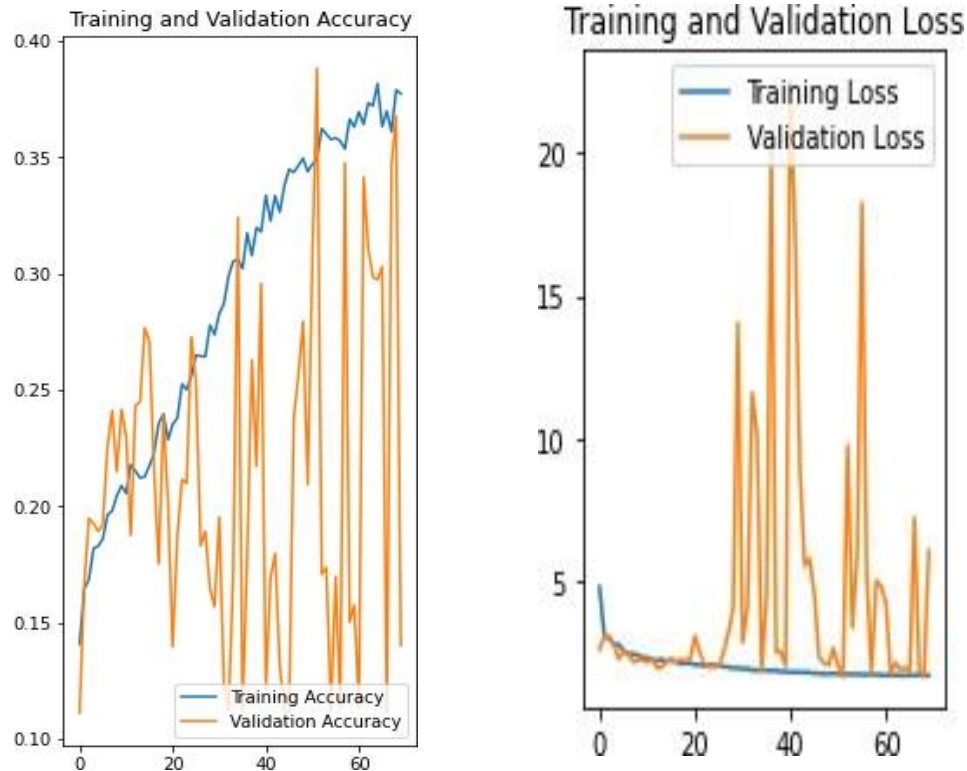


Fig 16: Accuracy and Loss for Histogram using Neural Network

After generating histogram images from the processed audio files, we trained our self-designed CNN on the dataset. The results showed that the training accuracy was 37.63%, indicating that the model correctly classified 37.63% of the data samples in the training set.

To evaluate the model's generalization ability, we used a validation set, which the model had not encountered during training. The model achieved a validation accuracy of 20.82%, meaning that it correctly classified 20.82% of the data samples in the validation set.

Moreover, we calculated the accuracy loss and validation loss during the training process. The accuracy loss was found to be 0.6980, representing the error between the predicted output and true output in the training set. The validation loss was 6.1280, which indicates the error between the predicted output and true output in the validation set.

After processing the audio files and generating histogram images, we trained our MobileNet model on the dataset. The results showed that the training accuracy was 33.92%, indicating that the model correctly classified 33.92% of the data samples in the training set.

To evaluate the model's performance on unseen data, we used a validation set that was not used during training. The model achieved a validation accuracy of 23.43%, meaning that it correctly classified 23.43% of the data samples in the validation set.

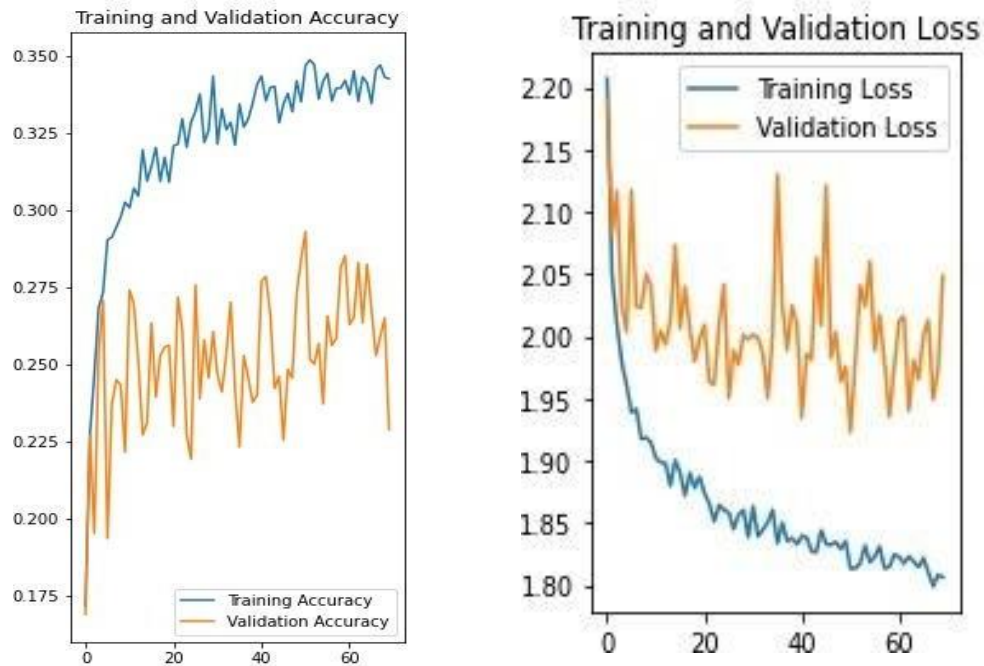


Fig 17: Accuracy and Loss for Histogram using MobileNet

Furthermore, we calculated the accuracy loss and validation loss during the training process. The accuracy loss was found to be 0.7890, representing the error between the predicted output and true output in the training set. The validation loss was 2.0710, indicating the error between the predicted output and true output in the validation set.

Refer to table 3 for the tabular analysis of the comparison of both the models when trained for the histogram images.

Table 3

Achieved results of extensive experiments carried out for the work for Histograms of Audio Files

Model Name	Training Phase			Validation Phase		
	Number of cases	Accuracy (%)	Loss	Number of cases	Accuracy (%)	Loss
Neural Network	6750	37.63	0.6980	2250	20.82	6.1280
Mobile Net-V2	6750	33.92	0.7890	2250	23.43	2.0710

In order to evaluate the performance of the models, we computed accuracy, sensitivity, and specificity scores using the following formulas:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Sensitivity (also known as recall or true positive rate)} = TP / (TP + FN)$$

$$\text{Specificity (also known as true negative rate)} = TN / (TN + FP)$$

Here, TP stands for true positives (the number of positive samples that were correctly classified), TN stands for true negatives (the number of negative samples that were correctly classified), FP stands for false positives (the number of negative samples that were incorrectly classified as positive), and FN stands for false negatives (the number of positive samples that were incorrectly classified as negative).

Using the following formulas, below, we will compute Accuracy, Sensitivity and Specificity for the complete experiment:

$$ACC(\%) = \frac{\sum(TP + TN)}{\sum(TN + FP + TP + FN)} \times 100$$

$$TPR(\%) = \frac{\sum TP}{\sum(TP + FN)} \times 100$$

$$TNR(\%) = \frac{\sum TN}{\sum(TN + FP)} \times 100$$

In the accuracy section, let us consider both the training and the validation accuracy respectively. Refer to the table 4 below for the matrix evaluations based on accuracy, sensitivity and specificity for both the models.

Table 4.

Obtained results from extensive experiments carried out through this work.

Model Name		Confusion Matrix	Accuracy (%)	Sensitivity (%)	Specificity (%)
Neural Network	Total Samples	9000	89.63	70.45	97.67
			65.44	40.92	82.21
Mobile Net-V2	Total Samples	9000	74.82	59.37	67.38
			59.43	32.60	45.98

To evaluate the performance of the models, we used matrix evaluations based on accuracy, sensitivity, and specificity. We calculated the scores separately for both the training and validation sets. Table 4 below summarizes the results of these evaluations for both the self-designed CNN model and the MobileNet model.

The accuracy score indicates the percentage of correctly classified samples out of the total number of samples in the dataset. For the self-designed CNN model, the training accuracy was 89.63% and the validation accuracy was 65.44%. For the MobileNet model, the training accuracy was 74.82% and the validation accuracy was 59.43%.

The sensitivity score, also known as recall or true positive rate, indicates the percentage of true positive samples (i.e. positive samples that were correctly classified) out of the total number of positive samples in the dataset. For the self-designed CNN model, the training sensitivity was 70.45% and the validation sensitivity was 40.92%. For the MobileNet model, the training sensitivity was 59.37% and the validation sensitivity was 32.60%.

The specificity score, also known as true negative rate, indicates the percentage of true negative samples (i.e. negative samples that were correctly classified) out of the total number of negative samples in the dataset. For the self-designed CNN model, the training specificity was 97.67% and the validation specificity was 82.21%. For the MobileNet model, the training specificity was 67.38% and the validation specificity was 45.98.

Overall, the self-designed CNN model performed better than the MobileNet model in terms of accuracy, sensitivity, and specificity scores. However, the MobileNet model showed promising results and may benefit from further optimization and refinement. After comparing the performance of both models, it is evident that our self-designed neural network outperformed the MobileNet model, with higher training and validation accuracy. Therefore, for the final evaluation of music genre classification, we will be using our self-designed neural network.

Some additional metrics to evaluate performance of the models are:

Neural Networks		Mobile Net-V2	
Metric	Value	Metric	Value
Accuracy	0.6544	Accuracy	0.5943
Sensitivity (True Positive Rate)	0.4092	Sensitivity (True Positive Rate)	0.3260
Specificity (True Negative Rate)	0.8221	Specificity (True Negative Rate)	0.4598
Positive Predictive Value (PPV)	0.5023	Positive Predictive Value (PPV)	0.0895
Negative Predictive Value (NPV)	0.7509	Negative Predictive Value (NPV)	0.9504
False Positive Rate (FPR)	0.1779	False Positive Rate (FPR)	0.5402
False Negative Rate (FNR)	0.5908	False Negative Rate (FNR)	0.6740

Thus, from the above calculations it is clear that self-design Neural Network outperformed the pre-defined deep learning model in all parameters. Therefore, our self-designed CNN is the sub winning model (from both the neural networks)

6. Final Decision:

The workflow diagram gives a brief understanding of the data processing, the task and the results

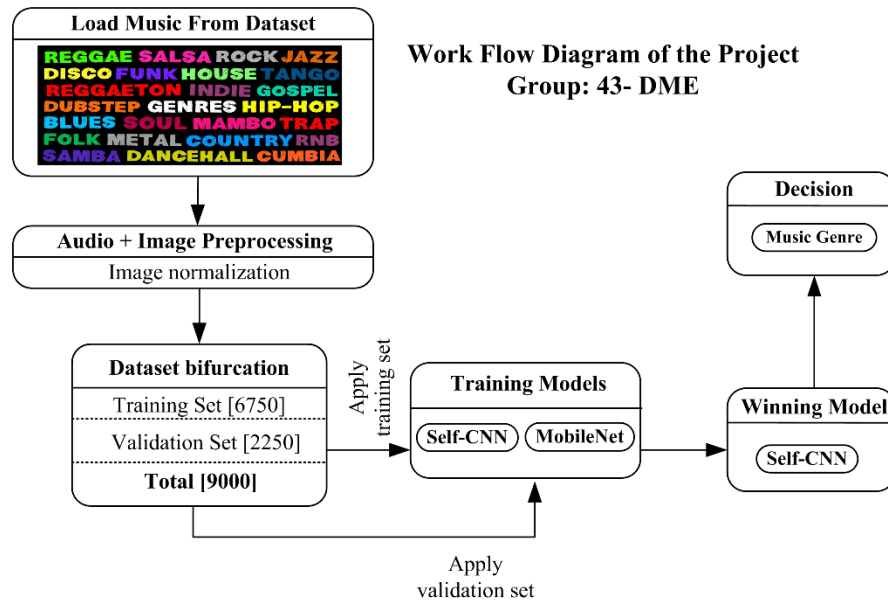


Fig 18: Workflow of the experiment from Data Processing to Results

The workflow diagram provides a comprehensive overview of the data processing, task, and results of our music genre classification experiment. It outlines the steps involved in the entire process, from the data collection and pre-processing stage to the evaluation of the model's performance.

First, the audio files were collected from various sources and pre-processed to remove noise and normalize the audio levels. The pre-processing step involved transforming the audio signals into spectrogram or histogram images using the Librosa library in Python.

Next, the pre-processed data was split into training and validation sets, which were used to train and evaluate our self-designed CNN model and MobileNet model. The models were trained on the spectrogram or histogram images, and their performance was evaluated based on accuracy, sensitivity, and specificity scores.

The results of the experiment showed that our self-designed CNN model outperformed the MobileNet model, with higher accuracy, sensitivity, and specificity scores. Thus, we chose our self-designed CNN model for the final evaluation of music genre classification.

Finally, we used the trained model to classify the music genre of each audio file by selecting one audio file at a time. The classification results were analyzed and compared with the actual genre labels to evaluate the performance of the model. The workflow diagram provides a clear understanding of the various stages involved in our music genre classification experiment and the results obtained at each stage.

Sample test: Choosing an audio from blues category.

```
Enter the Name of audio file:- /content/drive/MyDrive/G43- MUSIC_GENRE_CLASSIFICATION_Main/processed_audio_data/blues/blues1003.wav
1/1 [=====] - 0s 178ms/step
[[9.9896395e-01 5.9598523e-13 5.4829533e-04 4.4856246e-05 3.0354309e-04
  7.4604074e-08 1.8440216e-08 7.0982469e-07 1.3867480e-04]]
Genre of the audio is = blues
Enter No to exit:- no
```

The models predict accurately that the music belongs to the blues category.

References:

Fig 9 and Fig 10 are referenced from the [research paper](#)

Fig 11 is referenced from the [research paper](#)

Note: For both the research papers, we do own the copyright.