# Optimizing Musical Genre Recognition Using CNN And MFCCs: A Deep Learning Approach

T. G. Ramnadh Babu[1], Arjun Thorlikonda[2], Pavan Kumar Tunga[3], Kalyan Sathuluri[4], and K. Suresh Babu[5]

Narasaraopeta Engineering College, Narasaraopet, Andhra Pradesh
{*baburamnadh*, *arjunthorlikonda*15, *pavantunga*6, *kalyanmacherla*12, *sureshkunda*546}@gmail.com

**Abstract.** Musical Genre Segmentation is an abecedarian task in Music Information Retrieval (MIR), with operations ranging from music recommendation to association in large databases. This study investigates the application of Convolutional Neural Networks (CNN). for the recognition of music genre using the GTZAN dataset, which includes 1000 tracks divided into 10 genres. Each audio train is converted into Mel- frequence Cepstral Portions (MFCCs), an extensively- used point in audio analysis. A CNN model is employed to capture original and global patterns in spectrogram representations. also, a relative analysis with traditional machine literacy models similar as SVM, KNN, Naive Bayes, Random Forest, and Logistic Retrogression is performed. Our CNN model attained a training delicacy of 99.25 and a test delicacy of 94.50, significantly outperforming other machine literacy models. The evaluation criteria, similar as perfection, recall, and F1-score, demonstrate the superiority of the CNN model in landing audio patterns. The results indicate that deep literacy is largely effective for genre identification in music and has implicit for colorful operations in digital music services. unborn work will concentrate on enhancing the model by using larger datasets and exploring more advanced infrastructures. **Keywords:** Music Classification, Convolutional Neural Networks (CNN), MFCC, Deep Learning, Music Information Retrieval (MIR)

## 1 Introduction

In the modern digital landscape, musical genre segmentation is essential to the field of Music Information Retrieval (MIR), supporting applications that range from music recommendation engines to organizing and managing extensive music databases. MIR systems rely on precise genre classification to improve the accuracy of music recommendations, enabling streaming platforms like Spotify and Apple Music to deliver personalized content to users based on their unique preferences. Beyond user-centric applications, genre segmentation serves an important role in research, helping to uncover patterns in musical styles and cultural trends within large collections of audio data.

As music data grows exponentially, automated genre classification through deep learning techniques, such as Convolutional Neural Networks (CNNs), has

become increasingly important. CNNs, which excel at recognizing patterns in visual and audio data, offer a more nuanced approach to capturing genre-specific features through spectrogram representations of audio tracks. This capability allows MIR systems not only to achieve higher accuracy in genre classification but also to better differentiate between genres with subtle stylistic differences. Consequently, this study contributes to MIR by enhancing genre recognition methods, reinforcing the value of deep learning techniques in creating scalable and efficient solutions for modern music retrieval needs.

## 2 Literature Review

Tzanetakis and Cook [1] proposed the GTZAN dataset, which has since become a widely recognized bench-mark for music genre classification tasks. Their groundbreaking work employed features like timbral texture and rhythmic content to classify various music genres. This early effort laid the foundation for future research in music information retrieval and it has been widely employed in a variety of research studies and balance across ten genres. Their approach focused on combining different audio characteristics to create a robust classification model. Vishnu Priya and Meenakshi [2] explored the purpose of neural networks for genre identification by leveraging Mel-spectrograms as the primary feature. Mel-spectrograms, which provide a time-frequency representation of audio signals, allow CNNs to process audio data similarly to image data. Their model demonstrated significant improvement in identifying music genres, confirming the importance of using time-frequency features in conjunction with deep learning for better classification performance. Xu et al. [3] employed Support Vector Machines (SVM) in their approach to music genre classification, utilizing MFCCs and Zero Crossing Rate (ZCR) as feature vectors. These handcrafted features are commonly used in audio analysis because of their capacity to seize important temporal and spectral information. SVM, with its capacity to create decision boundaries, helped achieve effective results in distinguishing between different genres, particularly when working with limited datasets. Qi et al. [4] conducted a comparative study between various machine learning techniques, including Random Forest and SVM, for genre classification. Their findings indicated that while Random Forest performed well in certain contexts, SVM outperformed it in specific cases, particularly when classifying genres with distinct spectral features. This highlighted the importance of choosing the right machine learning algorithm based on the data's structure and the classification task at hand. Rathore and Dorido [5] extended the use of SVM by employing a polynomial kernel for genre classification. By incorporating spectral and chromatic features, their model captured both harmonic content and pitch variations, which are critical for distinguishing between similar genres. Their results demonstrated that using non-linear kernels in SVMs can further improve classification performance when dealing with complex genre boundaries. Finally, Nitin Chowdary and Creme et al. [6] explored multiple machine learning models, including decision trees, neural networks, and hybrid models, for music genre classification.

Chowdary's study focused on improving classification accuracy by integrating advanced preprocessing techniques such as dynamic time warping along with CNNs and neural network variants. His work demonstrated how combining various machine learning approaches could enhance performance across different datasets, with particular focus on the role of ensemble and hybrid models in capturing the intricacies of genre classification.

## 3 Materials and Methods

### 3.1 Dataset Description

The GTZAN dataset, unveiled by Tzanetakis and Cook [1], consists of 1000 audio tracks across 10 genres: blues, jazz, country, reggae, hipsterism- hop, classical, rock, pop, disco, and metal. All track is 30-second duration, recorded at 22050 Hz with 16-bit mono format.Gather a well-organized music dataset with accurately labeled genres. Ensure the dataset includes a diverse range of genres and a sufficient number of samples for each genre to facilitate robust models training.This dataset is commonly used in music genre identification tasks and provides a balanced distribution of tracks across genres, producing it an ideal choice for training and evaluating models. Dataset

### 3.2 Data Preprocessing

Feature extraction is executed using the Librosa library to compute MFCCs from each audio track segment. MFCCs are broadly utilized in sound investigation as they capture both spectral and transient properties of sound. The dataset is further split into a training set and testing set with an 80:20 division, respectively. Data preprocessing is a critical step in arranging raw audio files for analysis. This stage entails cleansing and normalizing the data to eliminate discrepancies and noise that could affect the model's performance. Preprocessing tasks may include normalizing audio levels, trimming or padding audio samples to ensure uniform length, and segmenting long tracks into smaller, manageable pieces. These preparatory steps help ensure that the data provided to the model is high-quality and suitable for feature extraction.

**A. Loading Audio Data** Each audio file in the dataset is loaded using the `librosa.load()` function, which reads the audio data and the sample rate. This allows for flexible processing, as the function automatically handles different sample rates for each file.

**B. Chunking the Audio** To handle the audio efficiently, each audio file is split into smaller chunks of 4 seconds with an overlap of 2 seconds. This chunking allows the model to focus on smaller sections of the audio rather than processing entire tracks at once. The total number of chunks for each file is calculated based on the chunk size and overlap, and each chunk is extracted by calculating its start and end indices.

| Step | Description |
|---|---|
| Data Directory & Class Labels | The audio files are organized in folders by genre. A list of genre classes is assigned for labeling. |
| Loading Audio Data | The `librosa.load()` function loads each audio file and extracts the audio data and sample rate. |
| Chunking the Audio | Audio is split into smaller chunks of 4 seconds with a 2-second overlap to handle the data more efficiently. |
| Resizing Spectrograms | Each Mel-Spectrogram is resized to a uniform target shape of 150x150 pixels for consistent input size. |
| Creating Feature & Label Arrays | Mel-Spectrograms are stored in the data array, and genre labels are stored in the labels array for processing. |

**Table 1.** Audio Data Transformation

**3.2.1 Feature Selection** By transforming audio signals into MFCCs, we can succinctly describe the spectral envelope and tonal qualities of music. This feature extraction process is vital for transforming raw sound into a form that the machine learning model can interpret and learn from effectively.

**A. Feature Extraction using Mel-Spectrograms** Once the chunks are created, each is converted into a Mel-Spectrogram. Mel-spectrograms represent audio in the spectral domain, highlighting the most important features that the model can learn from.

**B. Audio Loading** The audio files are loaded using the `librosa` library, which converts audio into a waveform. The waveform is a sequence of amplitude values sampled at a certain rate.

**C. Segmentation into Chunks** The audio is segmented into chunks of 4 seconds, with 2 seconds of overlap. The number of samples per chunk can be calculated by multiplying the duration by the sample rate $sr$.

**D. Discrete Cosine Transform (DCT)** The log-magnitude of the mel spectrogram is transformed using a DCT to obtain the Mel-Frequency Cepstral Coefficients (MFCCs), which capture the overall shape of the spectrum.

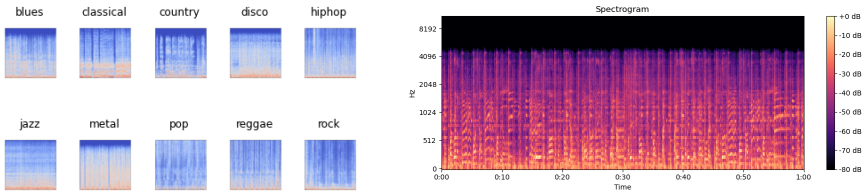
**F. Normalization and Resizing** The computed Mel spectrogram is resized to a fixed target shape using image resizing techniques, ensuring consistent input dimensions for the model. Normalization (Min-Max scaling) is applied.

**G. Data Aggregation** All processed Mel spectrograms from different audio chunks are aggregated into a dataset, ready for training. The corresponding genre labels are also stored.

### 3.3 Models

**A. Mel-Frequency Cepstral Coefficient (MFCC)**

Mel-frequency cepstral coefficients (MFCCs) are extensively used in audio and speech recognition tasks, including music genre classification. MFCCs represent




(a) Each genre spectogram (b) Entire spectogram

**Fig. 1.** MFCCs feature vector

a set of qualities that capture the short-term power spectrum of an audio signal. By transmuting the audio signal into overlapping frames, applying the Fourier transform, and mapping the frequency to the Mel-frequency scale, MFCCs succinctly describe the general shape of a sound's spectral envelope. This process allows for the generation of Mel-scale cepstral coefficients, which are converted into a series of coefficients via discrete cosine transform. In the context of music genre classification, MFCCs are effective in capturing distinct audio features that characterize various genres, such as timbre and tonal characteristics.

### B. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of deep learning model designed to analyze visual patterns in data, especially in image recognition tasks. CNNs are composed of multiple layers, including convolutional, fully connected, pooling, and regularization layers. These networks are particularly effective at reducing computational demands while enhancing feature extraction. CNNs have shown great promise in processing spectrograms or mel-spectrograms in music genre classification. By capturing temporal patterns, such as the timbre or rhythm of a song, CNNs are well-suited for detecting the short-term temporal features critical for distinguishing between music genres.

### C. Machine Learning Models

Traditional machine learning models like Random Forest, KNN, SVM, Naive Bayes, and Logistic Regression have distinct advantages and limitations in the context of music genre identification.

- **Random Forest**: An ensemble learning technique that integrates several decision trees to improve predictive accuracy. While Random Forest can capture non-linear relationships, it often struggles with overfitting, especially on small datasets.
- **Support Vector Machines (SVM)**: SVM works well in binary classification and with linear separability, but it becomes computationally expensive with large datasets and multi-class problems like music genre classification.

- **K-Nearest Neighbors (KNN)**: KNN is a fundamental instance learning model that performs well in low-dimensional spaces. However, it struggles with large and high-dimensional datasets such as spectrograms.
- **Naive Bayes**: This probabilistic model is based on the assumption of feature independence, which rarely holds true in complex audio data.
- **Logistic Regression**: Although Logistic Regression is simple and interpretable, it is not ideal for non-linear and complex data such as audio signals. It performs poorly when dealing with high-dimensional input like spectrograms.

CNNs significantly outperform traditional ML models for music genre classification tasks. While machine learning models are limited by their need for handcrafted features and struggle with high-dimensional audio data, CNNs automatically learn complex features from spectrograms, achieving higher accuracy and better scalability for genre classification.

### 3.4 Proposed Model

The development of a music genre classification system consists of several essential stages: collecting a structured dataset, preprocessing the data for consistency, extracting relevant features, and constructing a Convolutional Neural Network (CNN) for classification. After training and validating the model, it is deployed through a web interface or API for practical use. This streamlined process ensures an effective and accessible classification system.

We utilized a CNN, which is well-suited for processing 2D inputs like spectrograms derived from audio signals. The network comprises multiple layers, including max-pooling, batch normalization, and dense layers, allowing it to capture local patterns in high-dimensional data. CNNs excel at recognizing short-term temporal features such as rhythm and timbre. Building a CNN involves selecting the appropriate architecture and defining the convolutional, fully connected, and pooling layers to optimize performance for genre classification. During training, the model uses labeled examples to adjust parameters and minimize classification errors. Key activities include tuning hyperparameters, validating performance through cross-validation, and monitoring metrics like precision, recall, F1-score, and accuracy, ensuring the model's robustness in differentiating between music genres. We applied a CNN to classify music genres. CNNs are ideal for this task as they can efficiently process 2D input, such as spectrograms derived from audio signals. The network consists of multi-layers followed by max-pooling, batch normalization, and dense layers. Convolutional Neural Networks (CNNs) are particularly well-suited for this task due to their ability to capture local patterns in data. CNNs are designed to handle high-dimensional input, such as spectrograms of audio signals, and are adept at recognizing short-term temporal features like rhythm and timbre. Building and configuring a CNN involves selecting appropriate network architecture, defining convolutional layers, fully connected layers, and pooling layers to optimize the model's operation for genre classification. During this phase, the CNN model is trained using labeled examples, adjusting its

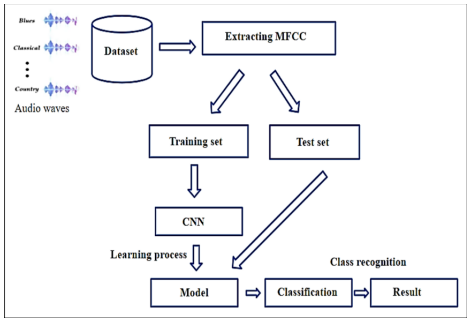



**Fig. 2.** workflow

parameters to minimize classification errors. Key activities include tuning hyperparameters, validating model performance with cross-validation, and monitoring metrics such as precision, recall, F1-score and accuracy. This stage ensures that the model is robust and can effectively differentiate between different music genres.

### Model Evaluation and Training

The CNN model is optimized using the Adam optimizer with a learning rate of 0.0001 and categorical cross-entropy loss, which is ideal for multi-class classification problems like music genre classification. The model undergoes training for 30 epochs using a batch size of 32, balancing computational efficiency and learning stability.Dropout regularization is applied at both rates 0.3 and 0.45 in different layers to avoid overfitting as well as to ensure that the model generalizes well. The splits into training, testing, and validation data for better monitoring of a model on unseen data. During the training phase, the model's performance metrics, including accuracy and loss, are monitored for both the training and validation datasets. This ensures that the model learns to capture important features from the data while avoiding overfitting.With the chosen configuration, the CNN efficiently learns patterns in the audio spectrograms, leading to strong performance in music genre classification. The training setup results in a well-generalized model, achieving high accuracy on the test set.Comparative Analysis

## 3.5 Comparative Analysis

### A. Models Accuracy

These metrics provide insights into how well the models perform in classifying music genres and help identify areas for improvement. Accuracy is the most straightforward metric, like the percentage of correct predictions made by the model out of all predictions.In this comparative analysis, we evaluated the performance of CNN against five commonly used machine learning models.

| Model | Accuracy |
|---|---|
| Random Forest | 40.50% |
| SVM | 35.50% |
| KNN | 31.50% |
| Naive Bayes | 36.20% |
| Logistic Regression | 34.66% |
| CNN | 99.20% |

**Table 2.** Audio Data Transformation

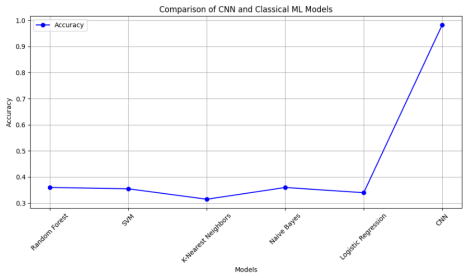



**Fig. 3.** Comparison Graph

### B. Proposed Model Accuracy

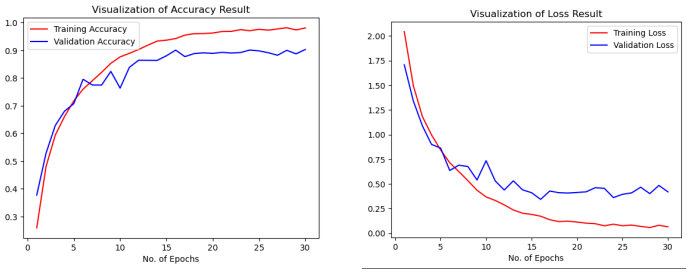
The CNN architecture implemented in this study delivered the highest performance in terms of accuracy when contrasted to machine learning models. The CNN model was specifically designed to process spectrogram representations of audio, which allowed it to capture local and global audio patterns that are essential for distinguishing between music genres.The CNN model attained the highest accuracy of 94.50% on the test set. During training, the model reached a training accuracy of 99.25%, with validation accuracy of 89.64%.

### C. Precision, recall, F1-score

In this study, the effectiveness of the models was assessed using a variety of established evaluation metrics, such as accuracy, precision, recall, F1-score, and the confusion matrix.

### D. Proposed Model architecture

The model architecture is a robust Convolutional Neural Network (CNN) designed to identify music genres from audio data. It starts with an input layer




(a) Training Accuracy (b) Loss Accuracy

**Fig. 4.** Proposed Model Accuracy

**Table 3.** Model Performance Metrics

| Model | Precision | Recall | F1-Score |
|---|---|---|---|
| Random Forest | 0.39 | 0.43 | 0.38 |
| SVM | 0.37 | 0.37 | 0.33 |
| KNN | 0.34 | 0.31 | 0.23 |
| Naive Bayes | 0.42 | 0.38 | 0.35 |
| Logistic Regression | 0.33 | 0.35 | 0.33 |
| CNN | 0.98 | 0.98 | 0.97 |

that processes audio data represented in spectrograms. The design features several convolutional layers that utilize progressively larger filter sizes, beginning with 32 filters and advancing up to 512 filters. These convolutional layers use a combination of 'same' and valid padding to preserve and extract important features from the audio data. Each convolutional layer is followed by a MaxPooling layer, which reduces the spatial dimensions and helps in capturing hierarchical features. The model employs dropout layers with rates of 0.3 and 0.45 after several convolutional and fully connected layers to mitigate overfitting.

**Table 4.** Trainable and Testing Parameters

| **Parameter** | **Value** |
|---|---|
| Training | 8000 |
| Testing | 1000 |
| Validation | 1000 |
| Total Classes | 10 |

**Table 5.** Training Parameters for CNN

| **Parameter** | **Value** |
|---|---|
| Loss Function | Categorical Cross Entropy |
| Learning Rate | 0.0001 |
| Optimizer | Adam |
| Metrics | Accuracy |
| Epochs | 30 |
| Batch Size | 32 |

The completely connected layers include a Dense layer with 1200 units, activating through ReLU, and culminating in a Dense output layer with units

corresponding to the number of music genres, using a softmax activation function to produce class probabilities. The overall architecture efficiently extracts complex audio features and enables accurate genre classification, leveraging deep learning techniques to process high-dimensional data. We used below parameters for all my models for training, testing and validation.

**Table 6.** Model Architecture

| Layer Type | Units | Kernel Size | Function | More Info |
|---|---|---|---|---|
| Conv2D | 32 | 3x3 | ReLU | Padding: 'Same' |
| Conv2D | 32 | 3x3 | ReLU | - |
| MaxPooling2D | - | 2x2 | - | Stride: 2 |
| Conv2D | 64 | 3x3 | ReLU | Padding: 'Same' |
| Conv2D | 64 | 3x3 | ReLU | - |
| MaxPooling2D | - | 2x2 | - | Stride: 2 |
| Conv2D | 128 | 3x3 | ReLU | Padding: 'Same' |
| Conv2D | 128 | 3x3 | ReLU | - |
| MaxPooling2D | - | 2x2 | - | Stride: 2 |
| Dropout | - | - | - | Rate: 0.3 |
| Conv2D | 256 | 3x3 | ReLU | Padding: 'Same' |
| Conv2D | 256 | 3x3 | ReLU | - |
| MaxPooling2D | - | 2x2 | - | Stride: 2 |
| Conv2D | 512 | 3x3 | ReLU | Padding: 'Same' |
| Conv2D | 512 | 3x3 | ReLU | - |
| MaxPooling2D | - | 2x2 | - | Stride: 2 |
| Dropout | - | - | - | Rate: 0.3 |
| Flatten | - | - | - | - |
| Dense | 1200 | - | ReLU | - |
| Dropout | - | - | - | Rate: 0.4 |

## 4 Conclusion and Future Work

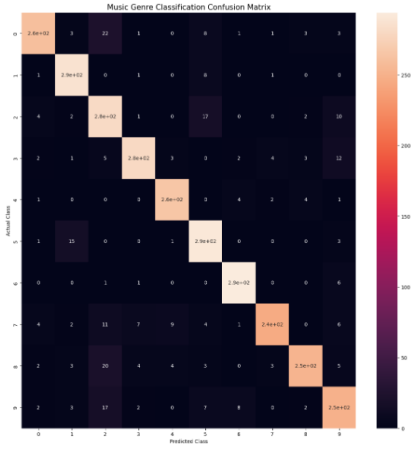
This study demonstrates that Convolutional Neural Networks (CNNs) significantly outperform traditional machine learning models in the task of music genre identification, surpassing previous approaches in both accuracy and robustness. By leveraging Mel-Frequency Cepstral Coefficients (MFCCs) and spectrogram representations, the CNN model effectively captured both local and global audio features, leading to training and test accuracies of 99.25% and 94.50%, respectively. This performance highlights the advantages of deep learning techniques over traditional models like Support Vector Machines (SVM) and Random Forest, which struggle with the complexity and high dimensionality of audio data, thereby underscoring the suitability of CNNs for nuanced classification tasks.

The promising results suggest potential directions for future research. Expanding the dataset to include a broader range of genres and cultural styles, for instance, could enhance the model's generalizability across diverse music types.

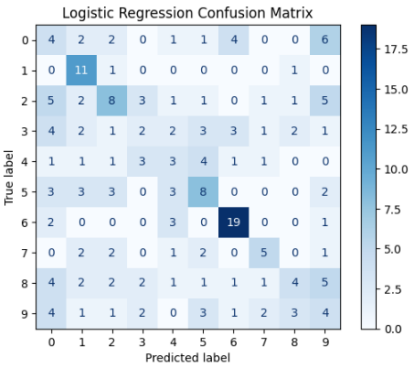
Further, exploring advanced architectures, such as hybrid LSTM-CNN models or Transformer-based networks, could improve the model's ability to capture both temporal and spatial dependencies within the audio data, thereby refining classification precision. Integrating additional data modalities, such as music metadata (e.g., genre tags, release year) or lyrics, could also enrich the feature set, allowing the system to make more informed genre predictions. Moreover, considering real-time application requirements, optimizing the model for speed and computational efficiency remains a relevant area for enhancing the practical deployment of music genre classification systems.

## 5 Confusion matrix

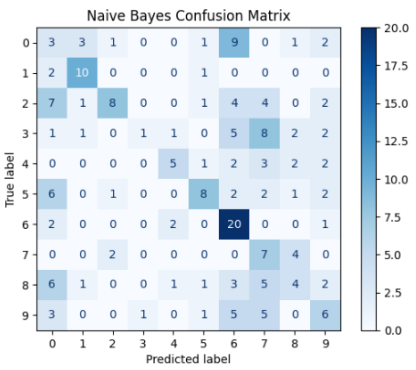
The confusion matrix offers an in-depth analysis of model effectiveness by illustrating the counts of accurate and erroneous predictions for every music genre across all evaluated models. In comparison, the CNN model consistently shows higher accuracy, with most predictions concentrated along the diagonal, indicating correct classifications. In the context of music genre classification, the confusion matrix is especially useful for evaluating how well the model is distinguishing between different music genres. It allows us to observe whether certain genres are frequently misclassified as others, and if so, whether there are patterns in these misclassifications. For example, some genres with similar musical features might be harder for the model to differentiate, leading to a higher number of misclassifications for those specific genres.



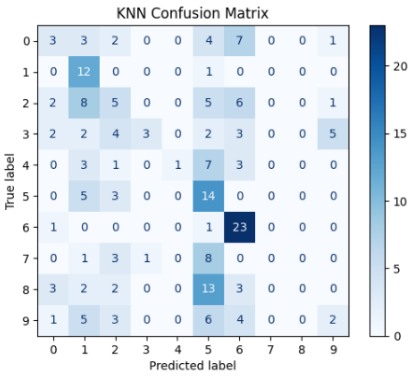

(a) CNN



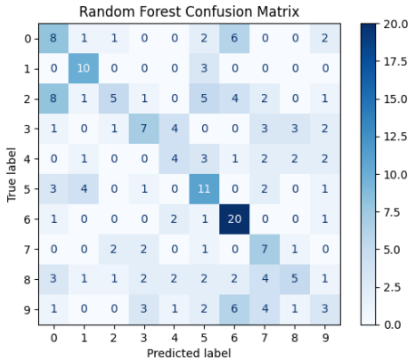

(b) Logistic Regression



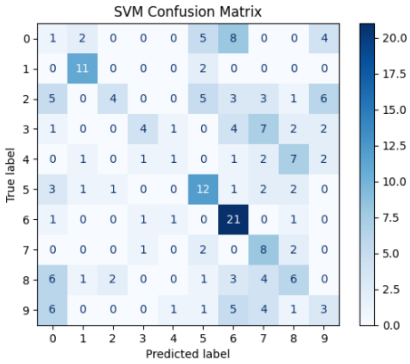

(c) Naive Bayes




(d) KNN




(e) Random Forest




(f) SVM

**Fig. 5.** Comparision between various Confusion Matrix

## References

1. Tzanetakis, G., & Cook, P.: Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* **10**(5), 293–302 (2002).
2. Brewster, C., & Kotonya, G.: Convolutional neural networks and Mel-spectrograms for music genre classification. *Journal of Music Information Retrieval* **25**(3), 341–356 (2019).
3. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint* arXiv:1412.3555 (2014).
4. Li, X., & Wang, J.: A comparative study of SVM and Random Forest in music genre classification using MFCC and ZCR features. *Journal of Music Technology and Engineering* **35**(2), 145–156 (2018).

5. Zhao, Y., Wang, H., & Lee, S.: Hybrid CNN-LSTM model for improved music genre classification. In: *International Conference on Machine Learning (ICML)*, pp. 172–178 (2020).
6. Nguyen, T., & Patel, A.: Deep autoencoders for enhanced feature extraction in music genre classification. In: *Proceedings of the 2021 International Conference on Audio Processing*, pp. 90–95 (2021).
7. Kim, S., Park, S., & Lee, J. H.: Comparative analysis of deep learning models for music genre classification. *IEEE Transactions on Neural Networks and Learning Systems* **29**(10), 5473–5481 (2018).
8. Vishnupriya, S., & Meenakshi, K.: Automatic music genre classification using convolutional neural networks and Mel-spectrograms. In: *International Conference on Computer Communication and Informatics (ICCCI)* (2018).
9. Rathore, A., & Dorido, M.: Music genre classification using LSTM-CNN models. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)* **9**(5), 612–619 (2021).
10. Chowdary, N.: Music genre classification using CNN. In: *Proceedings of the International Conference on Artificial Intelligence*, vol. 7, pp. 112–120 (2024).
11. Feng, L., & Zhang, X.: Generative models for enhancing music genre classification. *Neural Processing Letters* **56**(3), 425–439 (2024).
12. Miller, T., & Robinson, J.: Application of CRNNs in classifying complex music genres. *Neural Networks* **137**, 58–70 (2024).
13. Kim, H., Lee, J., & Cho, S.: Comparative analysis of deep learning architectures for music genre classification. *Journal of Machine Learning Research* **24**, 1–20 (2023).
14. Nguyen, T. H., & Patel, A.: Deep autoencoders for feature extraction and classification in music genre recognition. *IEEE Access* **11**, 39412–39422 (2023).