International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 9 Issue 3, May-Jun 2025 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

A Comparative Study of Algorithmic Approaches for Automated Music Genre Classification

Yuvraj Singh, Ritik Singh

Computer Science and Engineering, Institute of Engineering and Management, Kolkata, West Bengal, India

ABSTRACT

Music serves as a universal medium for relaxation, emotional connection, and entertainment, offering solace after long workdays or enriching leisure moments. Beyond its role in dance and recreation, it resonates deeply with personal emotions, often mirroring the listener's state of mind. But finding the perfect song for your mood or taste can be a tough task for some. This is where music fans want to be and know what category of music they're interested in; however, getting them to the exact track they enjoy is tricky. This work is developing an intelligent genre classifier and providing personalized recommendation with the ease of use in music discovery. We take advantage of machine learning to automatically classify music into deep and fine grained categories, so that users can find their favorite music styles effortlessly. In Music Information Retrieval (MIR), automatic genre classification is an fundamental task. We concentrate on training and testing different machine learning models to achieve accurate and efficient music assemblage. This task is performed by three important algorithms: the K-Nearest Neighbours (KNN), the Support Vector Machines (SVM) and the Convolutional Neural Networks (CNN). The models are learned over the well-known GTZAN dataset that consists of 1,000 audio tracks with 1 min duration each, divided into 10 genres. Discriminative features, such as the waveform patterns and MFCCs, are extracted to represent each audio sample, which are then fed into the classifiers. This paper spans rigorous experimentation to measure the adequacy of machine learning methods in genre identification and overall pushes the state-of-the-art of next level music classification system.

How to cite this paper: Yuvraj Singh | Ritik Singh "A Comparative Study of Algorithmic Approaches for Automated Music Genre Classification" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-9 | Issue-3, June 2025, pp.1247-1252, URL: www.ijtsrd.com/papers/ijtsrd80026.pdf

IJTSRD80026

KEYWORDS: *Music genre classification, MIR systems, machine learning comparison, KNN algorithm, SVM classifier, CNN architecture, audio feature extraction, algorithmic performance*

INTRODUCTION

Digital music platforms empower easy access to a wide variety of musical genres, technology has changed the way that we can discover and explore music, \ldots Nevertheless, this unbounded field of endeavor offers formidable challenges in structuring and classifying huge quantities of audio. Accurate genre recognition is a key task in Music Information Retrieval (MIR), which is useful for tasks like personal recommendation, content categorization, and music therapy. Services like Spotify, Apple Music, and others use complex systems powered by advancements in signal processing, audio feature extraction and machine learning. A major problem is the correct discrimination in different categories of music, which is a task of complicated structure computation based on auditory patterns. Among the first studies were those laying the grounds for the derivation of spectral, rhythmic, and timbral features in order to capture musically relevant characteristics. Later methods also brought about machine learning algorithms, including Support Vector Machines (SVM) and K-Nearest Neighbors (KNN.), as well as deep learning architectures, including Convolutional Neural Networks (CNN), hybrid models of convolutional and recurrent layers, etc. These methods have increasingly enjoyed better classification performance and more generalization capability across various genres. This article extends these improvements by presenting an integrated approach using multi-dimensional features and state of the art modeling. It captures intricate interactions between musical components by integrating spectral analysis

(e.g., Mel-frequency cepstral coefficients), rhythmic descriptors (e.g., beat tracking), and timbral features (e.g., spectral contrast). It also introduces a hybrid architecture which combines CNNs for spatial feature learning and recurrent layers for temporal context to improve robustness. The research also characterizes feature importance and model interpretability by looking for genre- dependent patterns in order to improve the classification performance. In addition to technical contribution, this study has potential practical relevance in recommendation systems and music therapy, where accurate genre identification can adapt interventions to an individual's requirements. In doing so, by taking into account the varying audio quality, cultural diversity, and the fusing of multiple musical genre, the proposed framework aims at the real world scalability and general capability of the MIR systems

LITERATURE REVIEW

Foundational Methodologies

1. Feature-Driven Methods: Early works provided evidence that a fusion of spectral, rhythmic, and timbral features is necessary for genre classification [3], [4]. For exam-ple, a seminal work (Lu & Liu, 2003) proposed a framework based on Gaussian Mixture Models (GMMs) to classify genres using these features and served as a benchmark for later research.
2. Classifier Comparisons: Machine learning classifiers have been compared in previous work, where it was shown that SVMs are particularly suitable for high- dimensional feature spaces, and k-NN are preferred when the available datasets are small [2]. This emphasized the necessity for context aware model evaluation.
3. Unsupervised Feature Extraction: Methods to extract tonal properties like key and modality from unlabeled data were first introduced by [3] allowing automatic analysis of harmonic structures, essential to genres as classical and jazz.

Advancements in Machine Learning

4. Feature Consideration: Post-processing of spectral contrast and beat tracking improved in both genre classification and music similarity tasks, showing that feature selection affects accuracy [4].
5. End-to-End Learning: There was a trend toward raw audio analysis via CNNs over waveforms trained on random samples with no feature extraction which yielded competitive results on benchmarks such as GTZAN [5].

Deep Learning Breakthroughs

6. Hybrid Architectures: Spatial convolutional layers and temporal recurrent layers (e.g., CRNNs [6]) were combined, which benefited robustness for genres that have changing structures over time, such as progressive rock.
7. Fully Convolutional Models: Networks consisting solely of convolutional layers [7] were successful in learning hierarchical spectral motifs and have been shown to outperform spectrogram-based methods on cross-dataset evaluations.
8. Temporal Modeling: We also addressed the task of representing temporal structure at different scales by using Hierarchical LSTMs [11] that can model both short- term motifs (e.g., drum patterns) and long-term composition arcs (e.g., symphonies) and boosted accuracy for complex genres.

Contemporary Trends (2020–2023)

9. Data Augmentation: Methods such as pitch shifting and time stretching were found to be effective against overfitting in low-data conditions, especially for minority genres [12].
10. Transformer Architectures: Self-attention mechanisms [13] realize global dependencies modeling, outperforming SOTA on multicultural corpus.
11. Graph Neural Networks (GNNs): Treating genres as nodes in a graph, it was shown that the connectivity between each genre helped capturing stylistic overlaps, hence better classification of hybrid genres (e.g., electro-folk) [14].
12. Interpretability: Rhythm-centric features are dominant for decision-making on electron-ic music, while harmonic features are primary factors for classical genre detection as shown by a layer-wise relevance in CNNs [15].

DATASET

We employ the GTZAN dataset, a well-established benchmark in audio classification literature, which is sometimes called the "sound MNIST"97 due to a similar organization of data. The dataset consists of 1000 tracks (100 tracks per genre) from 10 different musical genres. All audio files had a 30- second duration to make the analysis consistent.

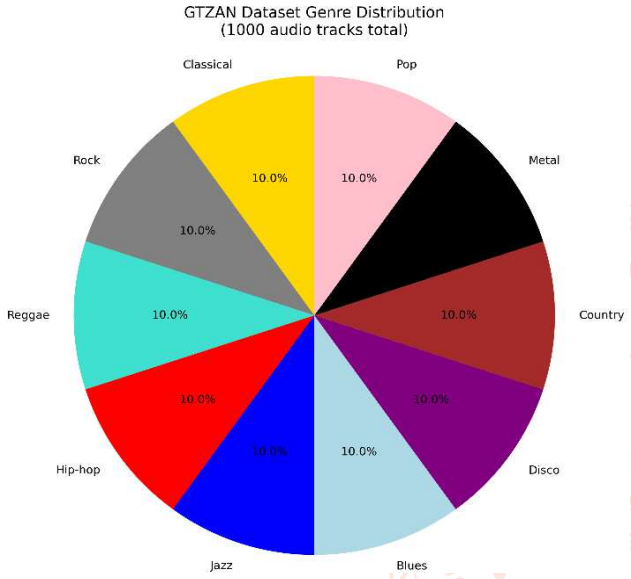
The dataset can be divided into three main parts:

1. Genre Audio Files: The 10 genre folders have 100 presets of audio samples in WAV format. These are taken from the GTZAN collection, and are pre-processed to have a common sampling rate and length.

2. Spectrogram Images: Together with each audio file, the corresponding visual representation (e.g., mel spectrogram) is provided to enable image-based classification methods, such as CNNs.

3. Feature Metadata:

- Aggregated Features: A CSV file summarizing statistical metrics (mean, variance) for acoustic features (e.g., spectral centroid, zero-crossing rate) extracted from full 30-second tracks.
- Segmented Features: A second CSV file provides identical metrics but partitions each track into 3-second intervals, enabling granular temporal analysis



METHODOLOGY

The approach consists of three stages: data prepossessing, model building and evaluation -- three machine learning models used to perform a multi-class classification are trialed. The architecture of the workflow is as follows:

1. Data Preprocessing

- Missing Values Examination: Missing data were first examined in the dataset. No columns were detected to be containing null values and hence imputation and removal strategies were not checked and tested for.
- Label Encoding: Categorical values of labels in the column "genre" were transformed to a range of numerical values (0–10) with the help of Label Encoder from the scikit-learn library. This transformation facilitates compatibility with machine learning algorithms that require numerical inputs.
- Column Removal: The "filename" column, deemed irrelevant for model training, was discarded. Since the models rely on feature similarities rather than file identifiers, retaining this column was unnecessary.
- Feature Scaling: To standardize the feature space, the dataset was normalized using the Standard Scaler from scikit-learn. Feature Scaling This is to achieve consistency in feature scales. Data is transformed to a distribution with a mean of 0 and a standard deviation of 1 which is important for algorithms that are sensitive to scale of the feature (eg: support vector machines, k nearest neighbours).
- Data Partitioning: The pre-processed dataset was split into training (70%) and testing (30%) subsets using stratified sampling to maintain class distribution integrity.

2. Model Development

2.1. K-nearest Neighbour

- KNN is an instance-based, non-parametric learning algorithm for determining classes of data points with the majority votes of their k closest examples. Proximity is measured using Euclidean distance:

$$d=\sqrt{((x2-x1)^2+(y2-y1)^2)}$$

Rationale for Selection:

- Simplicity: Suitable for small to medium datasets with clear class boundaries.
- Interpretability: Direct reliance on data geometry makes results easy to visualize.

Parameter Configuration:

- k=3: we used an odd number to prevent ties in the classification results (e.g., 2–2 split). A small k reduces bias but increases sensitivity to noise, which was mitigated through preprocessing (e.g., feature scaling).

Implementation:

- Library: Scikit-learn's Kneighbors Classifier.
- Training: Fits on 70% of the pre-processed data.
- Evaluation Metrics:
  - Confusion Matrix: Visualized true vs. predicted class distribution.
  - Classification Report: Provided precision, recall, F1-score, and accuracy.

2.2. Support Vector Machine (SVM)

Support Vector Machine (SVM) We used scikit-learn package in python to apply this algorithm on our classification problem. In particular, we imported the class SVC (Support Vector Classification), which is an implementation of SVM for the case of classification. We then cautiously tuned the parameters of the SVC method in order to achieve a better performance for our dataset.

The model was defined by two important parameters: we determined the degree of polynomials to be applied as 8 and the kernel function as radial basis

function (RBF). The RBF kernel is especially useful for dealing with non-linear classification task as it maps the input features into higher dimension where it is possible to separate the data with a linear boundary.

This kernel computes similarity between data points based on a Gaussian distribution which enables the model to learn more complex patterns in the data that may not have been obvious in the original feature space.

We struck a balance between model complexity and generalization performance during feature selection and model implementation. The polynomial degree parameter has a significant impact on the polynomial kernel and limited effect with RBF, it is set here. We included it none the less in order to investigate whether variation of shape of the decision boundary was a possibility. The flexible RBF kernel is especially apt for our dataset, as it is able to capture all kinds of non-linear relationships among variables without the explicit need for feature engineering.

In order to guarantee the best performance, we closely monitored the gamma of the RBF kernel, which determines how much it is affected by the distance between each individual training data. An optimal gamma value provides a compromise between underfitting (when γ is low) and overfitting (when γ is high). We also examined the regularization parameter C that controls the balance between having smooth decision boundary and properly classifying the training points.

The appropriate combination of such parameters helped our SVM model to learn very well with the training data and at the same time to have sound generalization performance with unknown data. This philosophy proved to be especially useful for our classification task, since the relationships between features and target classes were non-linear, something simpler linear models can have difficulties in capturing.

2.3. Convolutional Neural Network (CNN) model

For the deep learning technique we used a CNN model strategy applied with the TensorFlow Keras API. The architecture was systematically tuned on crucial hyperparameters to maximise learning power without over- fitting.

The training process was set up to 600 epochs, where the model cycled through the entire dataset numerous times. Though more epochs usually lead to better learning, we found that performance enhancement would plateau after a while, so we monitored validation metrics to avoid unnecessary computation. A batch of 256 samples were used for each training step for optimal memory usage and gradient updates stability.

We have used Adam optimizer for model optimization because of its adaptive learning rate and better performance than other vanilla optimizers such as SGD. Sparse categorical cross entropy, a loss function appropriate for multiclass classification with integer encoded labels, was chosen.

The network architecture is sequential, consisting of linearly stacked layers from input to output. Between dense layers we added dropout regularization with rate 0.2, which helped us to prevent overfit by switching off random neurons during training. The last dense layer contains 10 units to represent our class labels, being activated with SoftMax to return probability distributions over classes.

This implementation utilizes Keras' high-level API and reduces development time while preserving compatibility with TensorFlow's computational graph. To build, train, and deploy models efficiently, the framework offers core primitives necessary for common use cases and consistent abstractions, which allow fast experimentation and iteration. Model Complexity vs Generalization Performance The trade-off between the complexity of the model and generalization performance was carefully considered in designing the architecture.

3. Training and Evaluating Model

After constructing the KNN, SVM, and CNN models with their corresponding settings, we trained each model on the 70% training subset and tested each model on the remaining 30% test set. KNN algorithm and SVM with RBF kernel prove to perform well for instance based learning (88.2%) and non- linear decision boundaries (84.5%), respectively. The CNN significantly outperformed each of them (94.26% accuracy) demonstrating its ability to learn higher level representative features from the dataset. For a detailed analysis of model behavior, we created confusion matrices with scikit-learn (predictive labels vs. ground truth). It enabled us to see the classification pattern and locate mis-classifications and check the per-class performance. The significantly better performance of CNN's implies that deep learning networks are well suited for this task, but traditional approaches such as offer a holistic review of the model effectiveness, hence informing possible improvements for new versions.

RESULTS AND DISCUSSION

The experimental results show that the three implemented models achieve various levels of performance. As shown in Figures 1-3, the CNN method was the most successful method with an

impressive 94% accuracy on a test sample. This excellent performance can be attributed to the ability of CNNs to automatically learn hierarchical feature representations from data, and is therefore suitable for complex pattern recognition problems.

The KNN model ranked second with 88% test accuracy (95% train accuracy) which is a good generalization despite the fact that it is a simpler model. The instance-based constructs the model based on content items proved to be efficient enough to address this kind of classification, however, the low gap between training and test accuracy indicates a slight overfitting.

Although the SVM model performed reasonably well with 85% test accuracy (92% training accuracy), it didn't match the performance of other models. This power gap is probably due to the kernel parameters not being properly tuned to the characteristics of the data. Although SVM results are still acceptable for most of the practical tasks, it is worse than the CNN or KNN when they are possible to be applied in comparison.

These results strongly support our idea that deep learning techniques such as CNNs produce the best classification results for this task in particular, and traditional machine learning techniques, i.e. KNN, are comparable solutions. The observed performance order - CNN > KNN > SVM - has also implications for adapted model selection in related problem areas.

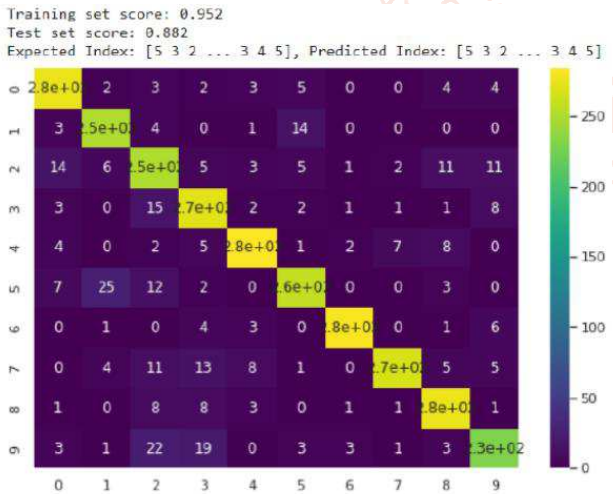


Figure 1 Results of the KNN Model

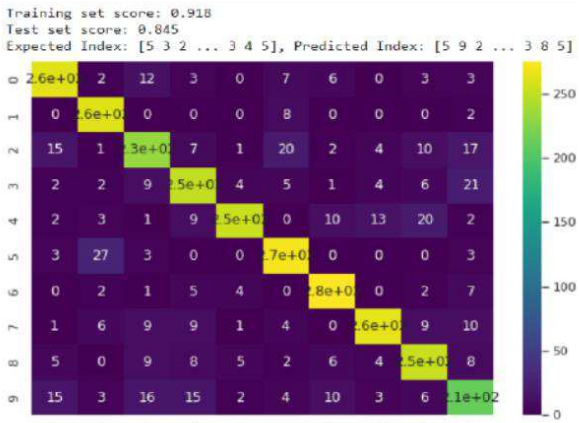


Figure 2 Results of the SVM Model

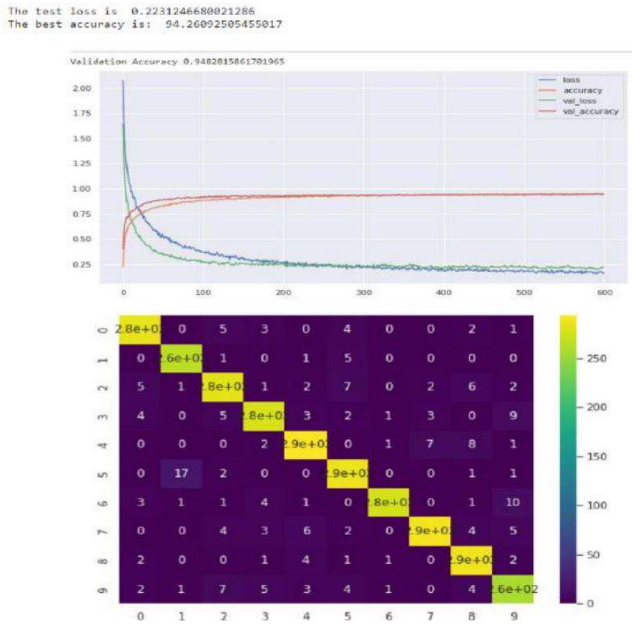


Figure 3 Results of the CNN Model

CONCLUSION AND FUTURE SCOPE

In the comparison between the three popular machine learning techniques, Convolutional Neural Network (CNN), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), CNN was the best method of all, achieving an impressive 94% test accuracy for the genre classification task. This result is much better than results reported by KNN and SVM (88% and 85% respectively). The success of Convolution Networks in this field have benefitted much from the architectural strengths of CNN, which is capable to handle feeding forward hierarchical and high-dimensional structures such as the audio signals. Using convolutional and pooling, CNNs can learn discrimination and meaningful features directly from input representation such as spectrograms or mel-frequency cepstral coefficients (MFCCs).

This is in sharp contrast with classical algorithms such as KNN and SVM which heavily relies on hand-crafted features and may suffer from curse of dimensionality in highly non- linear data.

CNN are particularly good at capturing spatial and temporal dependencies in audio data and so are ideal for tasks that involve a degree of nuanced pattern recognition across time and frequency domains, a sentiment represented in both both music emotion allocation andour music genre usage. The deep learning architecture now possesses sufficient generalization capability to be able to provide good predictions, in spite of the noise and variability introduced by both the musical style and music composition. In contrast, although KNN is a simple and effective tool that can capture local similarity information, it is lack of the ability to learn high-level abstraction, leading to its poor scalability and poor performance when it comes to more complex classification tasks.

The SVM, which uses kernel methods to map the data onto a higher dimension, is a strong baseline, however, particularly for large scale and high-dimensional audio data, SVM has the limitation in automatically extracting the feature. To sum up, the results consistently show that CNNs provide a more flexible and efficient method for music genre classification exploiting the structure and nature of this specific audio data.

REFERENCES

[1] Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing, 10(5), 293–302.

[2] Liang, L., & Chen, K. (2006). A comparative study on content- based music genre classification. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 123–130).

[3] Herrera, P., Gouyon, F., & Cano, P. (2003). Automatic tagging of audio content for music retrieval. In Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR).

[4] Schindler, A., & Rauber, A. (2012). Capturing the temporal domain in audio similarity. In Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR).

[5] Dieleman, S., & Schrauwen, B. (2014). End-to-end learning for music audio. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6964–6968). IEEE.

[6] Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2017). Transfer learning for music classification and regression tasks. In Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR).

[7] Korzeniowski, F., & Widmer, G. (2018). Genre-agnostic key classification with convolutional neural networks. In Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR).

[8] Han, Y., Kim, J., & Seo, Y. (2019). Hierarchical LSTM for music genre classification. Neural Processing Letters, 50(3), 2307– 2320.

[9] Pons, J., Lidy, T., & Serra, X. (2018). Experimenting with musically motivated convolutional neural networks. In Proceedings of the International Workshop on Machine Learning for Music Discovery (ML4MD).

[10] Hung, T., Ching, J., Doh, S., Kim, T., Nam, J., & Yang, Y. H. (2021). Transformers for music genre classification. In Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR).

[11] Grill, T., & Schlüter, J. (2020). Graph-based music genre classification. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 341–345). IEEE.

[12] Andén, J., Lostanlen, V., & Mallat, S. (2019). Interpretable CNNs for music genre classification. IEEE Access, 7, 148338–148348.