PROJECT REPORT

ON

HARMONIZING SOUNDS

Submitted By

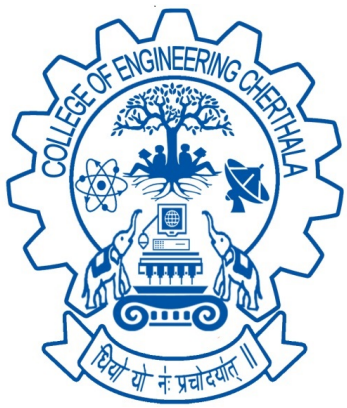
ADARSH P (CEC22MCA-2002)

under the esteemed guidance of

Mrs. Vishnupriya G S

Assistant Professor

Department Of Computer Science and Engineering



MAY 2024

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS
COLLEGE OF ENGINEERING, PALLIPPURAM P O, CHERTHALA,
ALAPPUZHA PIN: 688541,
PHONE: 0478 2553416, FAX: 0478 2552714
http://www.cectl.ac.in

PROJECT REPORT ON

HARMONIZING SOUNDS

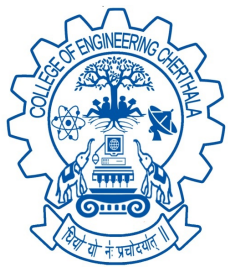
Submitted By

ADARSH P (CEC22MCA-2002)

under the esteemed guidance of

Mrs. Vishnupriya G S

In partial fulfillment of the requirements for the award of the degree
in
Master of Computer Application
of
APJ Abdul Kalam Technological University



MAY 2024

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS
COLLEGE OF ENGINEERING, PALLIPPURAM P O, CHERTHALA,
ALAPPUZHA PIN: 688541,
PHONE: 0478 2553416, FAX: 0478 2552714
http://www.cectl.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

COLLEGE OF ENGINEERING CHERTHALA

ALAPPUZHA-688541

C E R T I F I C A T E

This is to certify that, the project report titled **HARMONIZING SOUNDS** is a bonafide record of the **20MCA246 MAIN PROJECT** presented by **ADARSH P(CEC22MCA-2002)**, Fourth Semester MCA student, under our guidance and supervision, in partial fulfillment of the requirements for the award of the **MCA** degree of **APJ Abdul Kalam Technological University**.

Guide	**Co-ordinators**	**HOD**
Mrs. Vishnupriya G S	**Mrs. Janu R Panicker**	**Mr. Manilal D L**
Assistant Professor	Assistant Professor	Associate Professor
Dept. of Computer Science	Dept. of Computer Science	Dept. of Computer Science

ACKNOWLEDGEMENT

This work would not have been possible without the support of many people. First and the foremost, we give thanks to Almighty God who gave us the inner strength, resource and ability to complete our project successfully.

We would like to thank **Dr. Jaya V.L**, our Principal, who has provided with the best facilities and atmosphere for the project completion and presentation.We would also like to thank our HoD **Mr. Manilal D L** (Associate Professor, Department of Computer Engineering), our project coordinator **Mrs. Janu R Panicker** (Assistant Professor, Department of Computer Engineering), and our guide **Mrs. Vishnupriya G S** (Assistant Professor, Department of Computer Engineering) for the help extended and also for the encouragement and support given to us while doing the project.

We would like to thank my dear friends for extending their cooperation and encouragement throughout the project work, without which we would never have completed the project this well. Thank you all for your love and also for being very understanding.

ABSTRACT

The project aims to understand the complexities of sound through a multidimensional approach that includes visualization, classification, and comprehension. It uses advanced visualization techniques like spectrograms, waveforms, and frequency analysis to capture the rhythmic cadence, pitch variations, and tonal qualities of each audio snippet. Machine learning algorithms are used to classify sounds based on their distinctive features, allowing for systematic organization and interpretation. The project also uses advanced signal processing techniques to extract meaningful features from audio signals, which serve as the foundation for semantic interpretation and context-aware analysis. These features provide insights into the meaning and significance embedded within soundscapes. The project aims to foster a deeper understanding and appreciation of sound, paving the way for new avenues of inquiry in audio signal processing and beyond. The project's holistic exploration aims to illuminate the profound intricacies and nuances of auditory experiences.

Contents

List of Figures

Chapter 1

INTRODUCTION

Explores the world of music, focusing on its nuances and the intricate interplay of sounds, tones, and patterns. The GTZAN dataset, the MNIST of sounds in audio analysis, is used to classify 10 distinct genres, each curated with 100 audio files. The dataset provides a rich tapestry of musical diversity, spanning from classical to jazz to rock and electronic.

Mel Spectrograms, visual representations of these audio files, offer valuable insights into the underlying characteristics of each musical piece. Convolutional neural networks (CNNs) are used to tackle the task of genre classification. The journey involves meticulous experiments, feature extraction, model training, and performance evaluation.

The findings are applied to practical applications, such as recommendation systems and music organization. By developing personalized recommendation algorithms that cater to individual preferences and tastes, the report aims to streamline the process of music organization, facilitating seamless navigation and discovery within vast musical collections.

The ultimate goal is to enrich our appreciation of music's diversity and complexity, deepening our understanding of this universal language of human expression and forging new pathways for exploration and discovery in the captivating realm of sound.

Chapter 2

PROBLEM STATEMENT

2.1 Problem Statement

The GTZAN dataset is a valuable resource for audio comprehension and classification, offering a comprehensive analysis of 10 distinct genres and 100 audio files. Mel Spectrograms, visual representations of audio files, provide unique insights into the frequency content of audio signals, allowing researchers to extract valuable features for effective classification algorithms. Other attributes like tempo, rhythm, timbre, and harmony are also explored, transforming them into meaningful insights for sophisticated classification models.

The development of classification methods involves machine learning and deep learning techniques, including traditional classifiers like Support Vector Machines and Random Forests, as well as state-of-the-art neural network architectures like Convolutional Neural Networks and Recurrent Neural Networks. The task also involves model evaluation, validation, and optimization, requiring robust methodologies to ensure model reliability and generalization.

The GTZAN dataset is an invaluable resource for researchers and practitioners in audio signal processing and music information retrieval, offering a platform for advancing our understanding of music classification algorithms and their real-world applications. The task of comprehending audio files and devising effective classification methods requires meticulous attention to detail, rigorous experimentation, and innovative approaches.

2.2 Objective

Objectives of an harmonizing sounds are the process of understanding audio files through visualization, exploratory data analysis, genre classification using machine learning models, and developing a personalized music recommender system based on user preferences and audio features.

1. Understanding Audio Files: This objective entails a comprehensive exploration of the fundamental attributes of audio files. The project aims to delve into the intricate details of sound data, including its waveform, frequency spectrum, amplitude, and other relevant features. Through advanced visualization techniques such as spectrograms and waveforms, the goal is to gain a deeper understanding of the structural and perceptual characteristics of audio signals.

2. Exploratory Data Analysis (EDA): This objective involves conducting a thorough exploratory analysis of the dataset containing audio files. The project seeks to uncover insights into the distribution, characteristics, and patterns within the data. Through descriptive statistics, data visualization, and data profiling techniques, the aim is to identify trends, anomalies, and potential areas for further investigation.

3. Genres Classification: This objective focuses on utilizing machine learning models to classify audio files based on their genre. The project aims to leverage features extracted from provided CSV files containing statistical summaries of audio characteristics. By training and evaluating classification algorithms, the goal is to develop robust models capable of accurately categorizing audio files into different genres, thereby facilitating automated genre classification.

4. Recommender System: The objective here is to develop a recommender system capable of suggesting similar songs based on a given input. By leveraging techniques such as collaborative filtering or content-based filtering, the project aims to enhance music discovery and

exploration for users. The recommender system will analyze audio features and user preferences to generate personalized recommendations, thereby enriching the music listening experience.

Chapter 3

LITERATURE SURVEY

3.1 An Empirical Study on Structured Dichotomies in Music Genre Classification (2015)

The study delves into the realm of ensemble learning and dichotomy-based methods, aiming to enhance the genre classification process within music data. Through a meticulous comparison, binary classifiers such as Support Vector Machines (SVM), Naive Bayes (NB), k-Nearest Neighbors (k-NN), and Logistic Regression (LR) are scrutinized. The findings of the research reveal that dichotomy-based approaches fail to exhibit a significant enhancement in genre classification accuracy. In response to this, the study proposes heuristic methodologies for the construction of Nested Dichotomy Trees (NDTs), shedding light on innovative strategies to tackle this challenge.

In addition to presenting novel techniques, the study thoroughly discusses the intricacies and hurdles encountered in computational methods for genre classification in music. It emphasizes the importance of exploring various methodologies and structural frameworks to effectively navigate the complexities inherent in music data classification. The empirical experiments conducted throughout the study serve to validate the proposed approaches, providing valuable insights into their practical efficacy.

While the study highlights several advantages of its proposed methodologies, such as the comprehensive exploration of diverse methods and structures, it also acknowledges certain limitations. One notable limitation is the initial lack of accuracy observed in the proposed approach,

indicating the need for further refinement and optimization. Moreover, the study recognizes the significant performance gains achieved through the utilization of ensemble methods, underscoring the potential for further improvement in classification accuracy.

In conclusion, the study represents a significant contribution to the field of music genre classification, offering valuable insights into the effectiveness of ensemble learning and dichotomy-based methods. By identifying both the advantages and limitations of the proposed approaches, the study paves the way for future research endeavors aimed at addressing the remaining challenges and advancing the state-of-the-art in computational music genre classification.

3.2 Music Genre Classification and Recommendation by Using Machine Learning Techniques (2018)

The study employs digital signal processing techniques to extract various acoustic features from music audio samples. These features capture different aspects of the sound, such as pitch, timbre, and rhythm, providing a rich representation of the music. To analyze and classify these features effectively, the study employs advanced machine learning methods, including convolutional neural networks (CNNs) and deep learning architectures. These models are capable of automatically learning hierarchical representations of the input data, making them well-suited for tasks like music genre classification.

Among the machine learning algorithms evaluated in the study, the Support Vector Machine (SVM) algorithm emerges as the most successful in accurately classifying music genres. SVM is known for its ability to effectively handle high-dimensional data and nonlinear relationships between features, which is crucial for tasks like genre classification where the relationship between acoustic features and genre labels can be complex. By achieving the highest success rate in genre classification tasks, the SVM algorithm demonstrates its effectiveness in leveraging acoustic features to distinguish between different music genres.

However, despite the promising results achieved by the SVM algorithm, the study acknowledges several limitations that warrant consideration. Firstly, the focus solely on acoustic features overlooks other important aspects of music classification, such as lyrics or user preferences. Incorporating additional data modalities, such as text analysis for lyrics or user behavior analysis, could potentially enhance the accuracy and robustness of genre classification systems.

Furthermore, the evaluation of the proposed method is limited to the GTZAN dataset, a commonly used benchmark dataset in the field of music genre classification. While GTZAN provides a standardized platform for comparison, it may not fully represent the diversity of music genres and styles present in real-world scenarios. Therefore, conducting evaluations on additional datasets with a broader range of music genres would provide more comprehensive insights into the generalizability and performance of the proposed approach.

Additionally, the study lacks comprehensive comparisons with other state-of-the-art methods and algorithms in the field of music genre classification. While the SVM algorithm achieves promising results, comparing its performance against alternative techniques, such as random forests, k-nearest neighbors, or ensemble methods, would provide a more thorough understanding of its relative strengths and weaknesses. Moreover, exploring the impact of different feature representations, model architectures, and hyperparameter settings could help identify optimal configurations for genre classification tasks.

In summary, while the study demonstrates the effectiveness of SVM and digital signal processing techniques for music genre classification, addressing the mentioned limitations and conducting further investigations would contribute to advancing the state-of-the-art in this field. Integrating additional data modalities, evaluating on diverse datasets, and conducting comprehensive comparisons with alternative methods are essential steps towards developing more robust and accurate music genre classification systems.

3.3 A Hybrid Model For Music Genre Classification Using LSTM And SVM (2018)

The hybrid approach proposed in the study is undoubtedly promising, as it harnesses the strengths of both LSTM Neural Network and SVM classifiers to enhance prediction accuracy in music genre classification tasks. By training the classifiers separately and then amalgamating their results through the sum rule, the model showcases a sophisticated fusion of techniques. The utilization of the GTZAN music database, renowned for its diverse range of 1000 music files spanning ten genres, adds robustness to the model's training process. Extracting nine features from the database for training further enriches the model's understanding of musical nuances across different genres.

Despite its success in achieving an impressive 89% accuracy rate, the study leaves certain aspects unaddressed, notably the potential limitations and failure modes of the hybrid model. Every machine learning model, regardless of its complexity, is subject to certain constraints and failure scenarios. Without a thorough discussion on these, it's challenging to gauge the model's robustness in real-world applications. Additionally, the study overlooks considerations regarding computational resources. Training and deploying hybrid models, especially those combining deep learning and traditional machine learning algorithms, often require substantial computational power and memory resources. Understanding these requirements is crucial for practical implementation and scalability.

The study overlooks the trade-offs of selecting and combining different machine learning models, which have unique strengths and weaknesses. The integration process introduces additional complexities, such as LSTM Neural Networks excelling in temporal dependencies, and SVM classifiers handling high-dimensional feature spaces. Balancing these characteristics is crucial for optimal performance. While the hybrid approach shows promise in improving music genre classification accuracy, a more detailed discussion on potential limitations, failure modes, computational resource requirements, and model selection would provide a more nuanced understanding.

3.4 Exploring Data Augmentation to Improve Music Genre Classification with ConvNets (2018)

In the realm of music genre classification, the integration of Convolutional Neural Networks (CNNs) represents a significant stride towards enhancing accuracy and robustness. This study delves into the utilization of CNNs within the framework of music genre classification, particularly employing data augmentation techniques to further refine model performance. Leveraging the Latin Music Database (LMD) as the dataset substrate, the research meticulously explores various augmentation strategies, including but not limited to noise injection, pitch shifting, loudness variation, and time stretching.

One notable augmentation technique that stands out is the application of one-tone pitch shifting. This strategy emerges as a pivotal contributor to improved classification accuracy, demonstrating its efficacy in augmenting the dataset to better capture the diverse nuances inherent in music genres. By systematically manipulating the pitch of audio samples, the model gains a more comprehensive understanding of the underlying musical characteristics, thereby facilitating more precise genre classification.

The adoption of CNNs in this context brings forth a plethora of advantages, chief among them being the notable enhancements in accuracy. CNNs exhibit a remarkable ability to extract intricate patterns and features from raw audio data, thereby empowering the model to discern subtle genre-specific attributes with remarkable precision. Moreover, CNNs inherently possess superior generalization capabilities, enabling them to extrapolate learned patterns to previously unseen data instances, thereby bolstering the model's adaptability in real-world scenarios.

Furthermore, CNNs serve as a potent antidote to the perennial challenge of overfitting, a common pitfall encountered in machine learning endeavors. Through judicious architectural design and regularization techniques, CNNs mitigate the risk of overfitting, thereby ensuring that the model's learned representations encapsulate the essence of music genres without succumbing to spurious correlations within the training data.

However, amidst the plethora of advantages afforded by CNNs, it is imperative to acknowl-

edge and address certain caveats and drawbacks inherent in their utilization. One such concern pertains to the potential distortion of information introduced during the convolutional process. While CNNs excel at capturing hierarchical representations of audio features, there exists a risk of information loss or distortion at various abstraction levels, potentially undermining the model's fidelity to the original data.

Moreover, the adoption of CNNs invariably entails an escalation in computational complexity, particularly as the model architecture grows in depth and complexity to accommodate the intricate nuances of music genre classification. Consequently, this heightened computational burden necessitates substantial computational resources for training and inference, thereby posing logistical challenges, especially in resource-constrained environments.

Furthermore, the impact of fusion rules warrants careful consideration when employing CNNs for music genre classification. The integration of multiple sources of information, whether from diverse augmentation techniques or ensemble models, necessitates robust fusion mechanisms to effectively amalgamate disparate sources while preserving the discriminative power of individual classifiers. Failure to devise effective fusion rules may compromise the overall classification performance, underscoring the importance of meticulous algorithmic design and validation.

In summation, while CNNs undeniably represent a powerful tool for music genre classification, their efficacy is contingent upon judicious augmentation strategies, meticulous architectural design, and robust fusion mechanisms. By leveraging the strengths of CNNs while mitigating their inherent limitations, researchers can harness the full potential of deep learning in unraveling the intricate tapestry of musical genres with unparalleled accuracy and fidelity.

3.5 Neural Network Music Genre Classification (2019)

Music genre classification using machine learning techniques has emerged as a vibrant area of research, aiming to automate the process of categorizing songs into distinct genres. The study delves into a comprehensive analysis of various factors pivotal to the effectiveness of these techniques. Central to its investigation are the diverse song libraries employed, spanning across different musical styles and epochs, enabling a robust evaluation of the classifiers' generalization capabilities. Furthermore, the study meticulously explores an array of machine learning algorithms, encompassing supervised, unsupervised, semi-supervised, and reinforcement learning paradigms. Among these, particular attention is devoted to convolutional neural networks (CNNs), renowned for their prowess in discerning intricate patterns from complex datasets, especially in the realm of image and audio processing. Leveraging CNNs, the study underscores their efficacy in extracting critical features inherent to musical compositions, thereby facilitating accurate genre classification.

Moreover, the research underscores the paramount importance of dataset preparation, recognizing it as a fundamental determinant of model performance. Rigorous preprocessing, including normalization, feature extraction, and dimensionality reduction, is advocated to enhance the discriminative power of the classifiers. In tandem, meticulous attention is accorded to model architecture, with an emphasis on designing CNNs tailored to the intricacies of music data. Fine-tuning hyperparameters and incorporating regularization techniques are highlighted as indispensable strategies to mitigate overfitting and enhance model generalization.

The study highlights the potential of machine learning for music genre classification, but also highlights challenges such as designing and training neural networks, large labeled datasets, and the risk of overfitting. These issues require expertise in machine learning and music theory, as well as careful model validation and regularization to prevent overfitting. The study calls for further research and innovative methodological advancements to overcome these challenges and revolutionize music genre classification.

3.6 Time-Frequency Analysis for Music Genre Classification by using Wavelet Package Decompositions (2019)

The study employs wavelet package decomposition (WPD) as a novel approach for music genre classification, juxtaposed against the established method of Mel-Frequency Cepstral Coefficients (MFCC). The findings reveal notable enhancements in recognition rates when utilizing multiple singular values, underscoring the significance of factors such as WPD levels, sub-band selection, and singular value determination. One key advantage highlighted in the study is the discernible improvement in recognition rates achieved through WPD, showcasing its potential to outperform traditional MFCC methods. This improvement can be attributed to WPD's inherent multi-resolution characteristics, allowing for a more detailed analysis of the signal across different frequency bands. Additionally, WPD demonstrates efficacy in dimension reduction, which can streamline the classification process by eliminating redundant information and enhancing computational efficiency.

The study highlights the potential of Word Processing (WPD) as a viable alternative for music genre classification due to its ability to handle complex audio signals effectively. However, it also highlights several limitations and areas for further investigation. The lack of comprehensive discussions on the limitations of WPD could provide valuable insights into its applicability in real-world scenarios and its robustness in handling diverse music content. The study also lacks a thorough comparison of computational complexity between WPD and MFCC methods, which is crucial for assessing the practical feasibility of implementing WPD in resource-constrained environments. The absence of discussions on real-world implementation challenges implies a gap in understanding the practical implications of deploying WPD-based systems in production environments. Additionally, the study highlights the sensitivity of WPD to noise and variability in music content, which requires robust preprocessing techniques and model adaptations to enhance the resilience of WPD-based classification systems.

3.7 Utilizes Deep Learning CNN for Multilingual Music Genre Classification (2020)

The study presents a groundbreaking approach to music genre classification, leveraging the power of deep learning through convolutional neural networks (CNNs). By employing CNNs, the model attains an impressive classification accuracy of 93.3% across a wide range of languages, effectively mitigating biases that have plagued previous methodologies. This achievement is particularly significant as it not only demonstrates the efficacy of deep learning in music classification but also underscores the importance of addressing cultural diversity in dataset construction.

One of the strengths of the study lies in its utilization of a diverse dataset, which allows for robust model training and validation across different genres and linguistic contexts. Through the extraction of features from audio files using spectrograms and signal processing techniques, the model captures intricate patterns inherent in music across various genres, enabling it to make accurate genre predictions.

However, despite its high accuracy, the study acknowledges several limitations and challenges. One potential drawback is the possibility of misclassifications, wherein certain songs may defy conventional genre boundaries or possess characteristics that confound the model's classification scheme. Additionally, the selection of an appropriate algorithm poses a significant challenge, as different algorithms may exhibit varying degrees of effectiveness depending on the dataset and task at hand.

The study highlights a research gap in multilingual music genre classification, highlighting the need for refined methodologies and nuanced approaches. It emphasizes the complexity of cross-cultural music analysis and the need for significant computational resources and expertise for effective deep learning models. It also underscores the need for ongoing investment in computational infrastructure and interdisciplinary collaboration for future research.

3.8 Combined Transfer and Active Learning for High Accuracy Music Genre Classification Method (2021)

The study presents a novel approach to musical genre classification, departing from conventional methods such as Support Vector Machines (SVM) and Random Forests (RF). Instead, it employs a combination of Discrete Fourier Transform (DFT) and music attributes, leveraging transfer learning and active learning techniques to tackle complex scenarios in the classification process. By incorporating DFT, the method extracts essential frequency-domain features from audio signals, providing a robust representation of musical content. Additionally, the inclusion of music attributes enhances the classification process by capturing semantic information such as tempo, timbre, and rhythm patterns, contributing to a more comprehensive understanding of musical genres.

One of the notable advantages of the proposed method is its ability to achieve high accuracy in genre classification tasks. By harnessing transfer learning, the model can leverage knowledge from pre-trained models, potentially improving performance, especially in scenarios with limited labeled data. Moreover, the integration of active learning mechanisms allows the system to intelligently select the most informative instances for manual labeling, thereby reducing the burden of annotating large datasets while maximizing classification performance.

The method effectively classifies diverse musical styles, promoting inclusivity and preventing mainstream categories from dominating. It labels only 10-15% of unlabeled data, minimizing manual annotation and achieving satisfactory classification results. However, the study does not address specific disadvantages or limitations, suggesting further research. Future studies should investigate scalability, computational efficiency, generalization capabilities, and robustness to noisy or incomplete data. Examining the performance in dynamic or evolving music genres could provide valuable insights for refining the approach and advancing the field of musical genre classification.

3.9 A Hybrid Deep Learning Approach for Classification of Music Genres Using Wavelet and Spectrogram Analysis (2023)

The study introduces an innovative hybrid deep learning methodology designed to accurately classify music genres by leveraging both wavelet and spectrogram analysis techniques. In essence, this approach capitalizes on the strengths of these methods to extract meaningful features from audio data, thereby enhancing classification performance. Implemented in Python, the methodology harnesses several powerful libraries, including convolutional neural networks (CNNs), transfer learning-based techniques, multimodal training strategies, and hybrid models.

At its core, the methodology employs a combination of wavelet and spectrogram analysis to dissect the audio signals into their constituent frequency components and temporal patterns. By doing so, it captures both high-frequency details and broader spectral characteristics, thereby providing a comprehensive representation of the music data. This multi-level analysis enables the model to discern intricate nuances inherent in different music genres, leading to more accurate classification outcomes.

One notable aspect of the approach is its utilization of CNNs, which are renowned for their ability to automatically learn hierarchical representations of data. However, CNNs are prone to overfitting, wherein the model learns to memorize the training data rather than generalize patterns. To address this challenge, the study likely incorporates techniques such as regularization, dropout, or early stopping, which help prevent overfitting by imposing constraints on the model's complexity or training duration.

Furthermore, the methodology incorporates transfer learning, a technique wherein pre-trained models developed for one task are repurposed for another. By leveraging knowledge gained from tasks such as image classification, the model can expedite the learning process and potentially enhance classification accuracy, especially when training data is limited.

Moreover, the study employs multimodal training, a strategy that involves training the model on multiple types of data representations simultaneously. By combining wavelet and spectrogram features during training, the model can exploit complementary information encoded in each rep-

resentation, leading to improved robustness and generalization performance.

Ultimately, the hybrid model devised in this study surpasses the performance of other deep learning models in terms of accuracy, achieving impressive results of 81.5% and 71.1% accuracy on the GTZAN and Ballroom datasets, respectively. However, it is crucial to acknowledge the computational demands associated with such sophisticated models. The study likely conducts a careful analysis of computational resources and time requirements to ensure scalability and practical feasibility in real-world applications. Balancing computational efficiency with model performance is a crucial consideration for deploying the methodology in diverse settings, from academic research to commercial music recommendation systems.

3.10 Music Genre Classification Based on Fusing Audio and Lyric Information (2023)

The study introduces a novel approach to music genre classification, which integrates both audio and lyric data, thus presenting a holistic perspective on music analysis. By amalgamating these two distinct sources of information, the method aims to capture a more comprehensive representation of music genres, acknowledging the multifaceted nature of musical expression. To achieve this, the study employs advanced techniques such as BERT for text processing and Convolutional Neural Networks (CNNs) for extracting audio features.

One of the key contributions of this method lies in its exploration of various fusion strategies, which are essential for effectively integrating audio and lyric data. These fusion strategies encompass techniques such as feature concatenation, decision weighting, and a hybrid fusion approach, each offering unique advantages in terms of capturing the inherent characteristics of different music genres. Through meticulous experimentation and analysis, the study seeks to identify the most effective fusion method that optimally combines the strengths of both audio and lyric data.

Despite its promising potential, the method encounters several challenges that warrant careful consideration. One such challenge pertains to learning rate issues, which can significantly impact the convergence and optimization process of the neural network models. Addressing these issues requires fine-tuning the learning rate parameters and implementing adaptive learning algorithms to ensure stable and efficient training.

Moreover, the vectorization complexities associated with processing textual data, particularly with models like BERT, pose another obstacle that must be overcome. These complexities arise from the high-dimensional nature of word embeddings and the computational overhead required for processing large volumes of text data. Finding efficient vectorization techniques and optimizing computational resources are essential steps in mitigating these challenges.

Additionally, the study identifies the need for further exploration of fusion methods to enhance the overall performance and robustness of the classification system. While the proposed fusion strategies show promise, there is room for innovation and refinement to develop more so-

phisticated fusion techniques that better leverage the complementary nature of audio and lyric data.

In summary, the method proposed in the study offers a promising avenue for advancing music genre classification by leveraging both audio and lyric data. Through the integration of state-of-the-art techniques and the exploration of fusion strategies, the study contributes to a deeper understanding of music analysis while also highlighting important challenges and opportunities for future research in this domain.

Chapter 4

METHODOLOGIES

4.1 Audio Processing

4.1.1 Librosa Library

Librosa is a Python library designed for audio and music processing tasks, offering various functions and tools for analyzing and manipulating audio data. It helps break down complex audio files into manageable pieces, identifying essential elements like rhythm, melody, and texture. By utilizing Librosa, we can explore audio nuances and extract valuable insights for our recommendation system.

4.1.2 Feature extraction

Feature extraction is a crucial process in converting music into a format that computational algorithms can understand. It involves identifying key characteristics like tempo, pitch, timbre, and spectral in audio signals. These features form the basis of analysis, allowing for structured and quantifiable representation of complex musical information.

4.2 Convolutional Neural Network Learning Classification

4.2.1 Data Preparation

- Read the data from the features_3_sec.csv file: The first step involves loading the dataset from the provided CSV file named features3sec.csv. This dataset likely contains audio features extracted from audio files, along with corresponding labels indicating the genre or category of each audio sample.

- Extract features and labels: After loading the dataset, the features and labels need to be extracted. The features typically represent various characteristics or attributes of the audio files, such as spectral features, rhythm patterns, or other relevant information extracted using audio processing techniques. The labels indicate the target categories or classes that the CNN will learn to classify.

- Normalize the features and encode the labels: Before feeding the data into the CNN model, it's essential to preprocess the features. Normalization ensures that all feature values are on a similar scale, preventing certain features from dominating the learning process due to their larger magnitude. Additionally, encoding the labels converts categorical labels into numerical format, which is required for training machine learning models like CNNs.

4.2.2 Convolutional Neural Network Model

- Define a Convolutional Neural Network (CNN) model using TensorFlow/Keras: In this step, a CNN architecture is defined using TensorFlow/Keras. The architecture typically consists of convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply filters to the input data to extract relevant features, while pooling layers downsample the feature maps to reduce computational complexity. Fully connected layers are used for classification based on the extracted features.

- Compile the model with appropriate loss and optimization functions: After defining the

CNN architecture, the model needs to be compiled. This involves specifying the loss function, which measures the model's performance during training, and the optimization algorithm, which adjusts the model's parameters to minimize the loss function. Additionally, performance metrics such as accuracy may be specified to monitor the model's progress during training.

- Train the model on the training data: Once compiled, the CNN model is trained on the training dataset. During training, the model learns to extract relevant features from the input data and make predictions based on these features. The training process involves iteratively feeding batches of training data to the model and updating its parameters using backpropagation and gradient descent.

- Evaluate the model's performance on the test data: After training, the performance of the CNN model is evaluated using a separate test dataset that was not used during training. This evaluation provides an unbiased estimate of the model's performance on unseen data. The model's accuracy, loss, and other relevant metrics are typically computed during evaluation.

- Predict the labels for the test set and calculate classification metrics: Finally, the trained CNN model is used to predict the labels for the test dataset. These predicted labels are compared against the ground truth labels to calculate classification metrics such as confusion matrix and classification report. These metrics provide insights into the model's ability to correctly classify instances into their respective classes and identify any areas where the model may need improvement.

4.2.3 Long Short-Term Memory Model

Data Preparation

- Read the same data: The data used for the LSTM model is the same dataset used for the CNN model. This dataset is typically stored in a file format, such as CSV (Comma Separated Values), containing features extracted from audio files along with their corresponding labels.

The data is read from the file into a DataFrame or a similar data structure suitable for further processing.

- Normalize the features and encode the labels: Before feeding the data into the LSTM model, it is essential to preprocess the features and labels. This preprocessing often involves normalization and label encoding:

  Normalization: The features are scaled to ensure that all feature values lie within a similar range. Common normalization techniques include Min-Max scaling or Standard scaling, which transform the feature values to a predefined range or standard distribution, respectively. Normalization helps stabilize the training process and improves the convergence of the model.

  Label encoding: Since machine learning models typically operate on numerical data, categorical labels need to be encoded into numerical format. Label encoding assigns a unique integer to each category in the label set. This enables the model to understand and process the labels during training and inference.

By performing these data preparation steps, the dataset is transformed into a format suitable for training the LSTM model. This ensures that the model receives properly formatted input data and can effectively learn from it during the training process.

LSTM Model

In this subsection, we outline the steps involved in defining, compiling, training, and evaluating a Long Short-Term Memory (LSTM) model for the learning classification task:

1. **Define a Long Short-Term Memory (LSTM) model using TensorFlow/Keras:**

   - The LSTM model architecture is defined using TensorFlow/Keras API, which provides high-level abstractions for building neural networks.
   - The architecture typically consists of one or more LSTM layers followed by one or more fully connected (Dense) layers.

- The input shape of the data must be specified in the first layer of the model to ensure compatibility with the input data.
- The activation functions, number of units or neurons in each layer, and other hyperparameters are chosen based on the problem domain and experimentation.

2. **Compile the model:**

   - After defining the model architecture, it needs to be compiled with appropriate loss and optimization functions.
   - The loss function quantifies how well the model's predictions match the actual labels during training. For classification tasks, categorical cross-entropy is commonly used as the loss function.
   - The optimizer determines how the model's weights are updated during training to minimize the loss. Adam optimizer is a popular choice due to its adaptive learning rate.

3. **Train the LSTM model:**

   - The compiled model is trained on the training data using the `fit()` method.
   - During training, the model iteratively adjusts its weights based on the optimization algorithm and the calculated loss.
   - Training involves passing batches of input data through the network and updating the weights through backpropagation.
   - The number of epochs (iterations over the entire dataset) and batch size (number of samples processed in each iteration) are important hyperparameters that influence the training process.

4. **Evaluate the LSTM model:**

   - Once the model is trained, its performance is evaluated on a separate validation set or test set to assess its generalization ability.

- Evaluation metrics such as accuracy, precision, recall, and F1-score can be computed to quantify the model's performance.
- Additionally, visualizations such as confusion matrices and ROC curves can provide insights into the model's behavior across different classes.

By following these steps, we can develop and assess the effectiveness of an LSTM model for the learning classification task.

4.2.4 Recurrent Neural Network Model

RNN Model

In this subsection, we outline the steps involved in defining, compiling, training, and evaluating a Simple Recurrent Neural Network (RNN) model for the learning classification task:

1. **Define a Simple RNN model using TensorFlow/Keras:**
    - The RNN model architecture is defined using TensorFlow/Keras API, similar to the LSTM and CNN models.
    - The architecture typically consists of one or more SimpleRNN layers followed by one or more fully connected (Dense) layers.
    - As with other neural network architectures, the input shape of the data must be specified in the first layer of the model.
    - Hyperparameters such as activation functions, number of units or neurons in each layer, and dropout rates can be adjusted based on experimentation and domain knowledge.
2. **Compile, train, and evaluate the RNN model:**
    - After defining the RNN model architecture, it needs to be compiled with appropriate loss and optimization functions.
    - The compiled model is then trained on the training data using the `fit()` method, similar to the LSTM and CNN models.

- During training, the model iteratively adjusts its weights based on the optimization algorithm and the calculated loss.
- The model's performance is evaluated on a separate validation set or test set using evaluation metrics such as accuracy, precision, recall, and F1-score.
- Visualizations such as confusion matrices and ROC curves can also be generated to gain insights into the model's performance across different classes.

By following these steps, we can develop and assess the effectiveness of a Simple RNN model for the learning classification task, similar to the CNN and LSTM models.

4.3 Pretrained Model

In this section, we discuss the process of utilizing a pretrained model to predict the genre of a random audio file:

- **Utilize a predefined function to fetch a random audio file:**
  - A predefined function is used to retrieve a random audio file from the dataset directory. This function abstracts away the complexity of file handling and ensures consistency in selecting random samples for prediction.
  - The function may involve operations such as directory traversal, file selection, and path manipulation to identify and retrieve a random audio file.
- **Predict the genre of the audio file using a pre-trained model:**
  - Once the random audio file is obtained, it is passed as input to a pre-trained machine learning or deep learning model.
  - The pre-trained model has been previously trained on a large dataset and learned to recognize patterns or features indicative of different audio genres.
  - By feeding the audio file into the pre-trained model, it performs inference and outputs a prediction or classification result indicating the predicted genre of the audio.
  - The prediction is typically obtained in the form of a probability distribution over the possible genres, which can be further analyzed or interpreted.

This process enables the automated prediction of audio genres using a pretrained model, facilitating tasks such as music recommendation and genre classification.

4.4 Machine Learning Classification

4.4.1 Libraries and Data Preparation

- **Import necessary libraries for classification:**
  - This section discusses the importation of libraries for classification tasks and visualization in Python. The libraries include scikit-learn, a machine learning library that offers efficient tools for data mining and analysis, XGBoost, an optimized gradient boosting library, matplotlib, and seaborn. Scikit-learn includes classification algorithms like Naive Bayes, SVM, and decision trees. XGBoost is known for its efficiency, accuracy, and speed in handling large datasets. Matplotlib is a plotting library that creates static, interactive, and animated visualizations in Python. Seaborn is a statistical data visualization library that simplifies complex visualizations and supports features like categorical plotting, multi-plot grids, and color palettes. These libraries are crucial for importing classification algorithms, performing data preprocessing tasks, training and evaluating models, and visualizing results.
- **Read the features_3_sec.csv file:**
  - The features_3_sec.csv file contains audio features extracted from audio files. It is read using a pandas library function, specifying the file path and handling parameters like delimiter, header, encoding, and column names. The CSV file is stored in memory as a DataFrame, allowing access and manipulation using pandas' functions and methods. This dataset serves as the input for data preprocessing and modeling steps in our classification task, ensuring accurate and reliable data analysis.
- **Normalize features and split the data into training and testing sets:**
  - To prepare a dataset for machine learning, it is essential to normalize the extracted features and split the dataset into training and testing sets. Normalization ensures all

features are on the same scale, which is crucial for many algorithms. Common techniques include min-max scaling, which scales features to a fixed range, and standardization, which transforms features to have a mean of 0 and a standard deviation of 1. Standardization is particularly useful when features have different units or scales.

Splitting the dataset into training and testing sets allows for training models on one subset and evaluating their performance on another. Common ratios for splitting the dataset are 70%-30% or 80%-20%, with the larger portion used for training and the smaller portion for testing. This ensures that the model is trained on sufficient data while having a separate set of unseen data for evaluation.

4.4.2 Model Assessment

- **Define a function to assess the accuracy of a classification model:**
  - The development of a reusable function to evaluate the accuracy of classification models is crucial during the model assessment stage. This function should compare predicted labels with actual labels from the test set, calculate various evaluation metrics, and return or display the results. Common metrics include accuracy, precision, recall, and F1-score, which provide insights into the model's performance.

    The function takes the predicted and true labels from the test set as input parameters and calculates these metrics using appropriate formulas. These metrics provide insights into the model's ability to correctly classify instances of different classes and its balance between precision and recall.

    After computing the evaluation metrics, the function returns or displays the results to provide a comprehensive assessment of the model's performance, allowing stakeholders to understand its strengths and weaknesses and make informed decisions regarding its use. This streamlines the evaluation process and ensures consistent and thorough assessment of classification models across different datasets and experiments.

- **Assess the performance of various machine learning models:**

- The model assessment stage is crucial in identifying the best-performing machine learning model for a given dataset. This involves evaluating a range of classification algorithms, such as Naive Bayes, Stochastic Gradient Descent, K-Nearest Neighbors, Decision Trees, Random Forest, Support Vector Machine (SVM), Logistic Regression, Neuronal Networks, and XGBoost.

  Naive Bayes is a probabilistic classifier based on Bayes' theorem, while SGD is an optimization algorithm that updates model parameters to minimize loss function. KNN classifies instances based on the majority class among their nearest neighbors. Decision Trees are tree-like structures where each node represents a decision based on features, while Random Forest is an ensemble method that constructs multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. Support Vector Machine (SVM) is a supervised learning algorithm that finds the optimal hyperplane to separate classes in a high-dimensional space. Logistic Regression is a linear model used for binary classification, while neural networks are deep learning models capable of learning complex patterns from data. XGBoost is an optimized gradient boosting library that provides efficient implementation of gradient boosting algorithms.

  By evaluating various classification algorithms and comparing their performance metrics, the most suitable model can be identified, leading to more accurate predictions and better decision-making.

4.4.3 XGBoost Model

- **Choose XGBoost as the best performing model:**
  - Select XGBoost as the preferred classification model based on its superior performance compared to other algorithms assessed.
- **Train the XGBoost classifier on the training data:**

  - Utilize the XGBoost library to train a classifier on the training dataset, leveraging its gradient boosting framework to iteratively improve model performance.

- **Predict labels for the test set and evaluate accuracy:**

  - Use the trained XGBoost classifier to predict labels for the test dataset.
  - Evaluate the accuracy of the model by comparing the predicted labels with the ground truth labels from the test set.

- **Compute the confusion matrix to visualize model performance:**

  - Generate a confusion matrix to visualize the performance of the XGBoost model in classifying different classes.
  - The confusion matrix provides insights into the model's ability to correctly classify instances and identify any misclassifications or confusion between classes.

4.4.4 Feature Importance

- **Compute feature importance using permutation importance technique:**

  - Employ the permutation importance technique to assess the importance of features in the XGBoost model.
  - This technique involves shuffling individual features and measuring the impact on model performance, thereby quantifying the contribution of each feature to the model's predictive power.

- **Visualize feature importance scores:**

  - Visualize the feature importance scores obtained from the permutation importance technique using plots or charts.
  - Visualization aids in interpreting the relative importance of features and identifying the most influential predictors in the classification task.

This comprehensive approach to machine learning classification involves data preparation, model assessment, selection of the best-performing model, and analysis of feature importance to gain insights into the predictive factors driving classification outcomes.

The project utilizes Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs) as machine learning models. CNNs are used to uncover patterns and structures within audio data, analyzing spatial hierarchies and local patterns in multidimensional data. They classify and categorize audio samples based on their distinct features, enabling informed recommendations to users. LSTMs, on the other hand, specialize in capturing temporal dependencies and long-range dependencies within sequential data. They model the temporal dynamics of audio signals, such as the progression of melodies, rhythms, and chord sequences. By leveraging LSTMs, the project can understand how different segments of a song interact and evolve over time, creating more nuanced and context-aware recommendations.

Traditional classifiers, such as logistic regression, decision trees, and support vector machines, are also essential tools in the project. These algorithms offer simplicity and interpretability, providing robust solutions for tasks like genre classification, sentiment analysis, and mood detection. By leveraging both modern and classical machine learning techniques, the project can achieve a more comprehensive understanding of audio data and enhance the robustness of its recommendation system.

4.5 Similarity Scoring

4.5.1 Cosine Similarity

Cosine similarity serves as the cornerstone of our recommendation system, enabling us to quantify the similarity between pairs of audio samples. Conceptually, cosine similarity measures the cosine of the angle between two vectors, representing their degree of alignment in multidimensional space. In the context of audio processing, cosine similarity compares the feature vectors extracted from different songs to determine their similarity in terms of musical characteristics, such as tempo, timbre, and harmonic content. By computing cosine similarity scores, we can identify songs that share common traits and preferences, facilitating personalized recommendations for users based on their musical preferences and listening habits.

In essence, our project represents a harmonious fusion of advanced technology and human intuition, with the overarching goal of enhancing the music discovery and recommendation experience for users. By leveraging cutting-edge techniques in audio processing, machine learning, and similarity scoring, we aim to create a platform that resonates with users on a deeply human level, enriching their musical journey and fostering a deeper appreciation for the art of sound.

Chapter 5

PROPOSED SYSTEM

5.1 Solution

The proposed system employs cutting-edge techniques to delve into the multifaceted nature of sound, from visualization to genre classification. Through meticulous data collection and feature extraction, we capture the essence of audio, including spectral features, rhythm patterns, and timbral characteristics. Leveraging a variety of machine learning models such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, Recurrent Neural Networks (RNN), XGBoost, and Pre-trained Models, we classify song genres with precision. Evaluation metrics ensure the robustness of our models, while cross-validation minimizes overfitting. Through systematic analysis, our system not only enhances understanding but also fosters a deeper appreciation for the richness of sound.

1. Comprehensive Data Collection: By gathering a diverse dataset representing various song genres, the system ensures comprehensive coverage, facilitating a thorough exploration of sound intricacies across different musical styles.

2. Advanced Feature Extraction Techniques: Through sophisticated feature extraction methods, the system distills complex audio signals into meaningful representations, including spectral features, rhythm patterns, and timbral attributes.

3. Machine Learning Models: We employ a variety of machine learning models such as Convo-

lutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, Recurrent Neural Networks (RNN), XGBoost, and Pre-trained Models to classify song genres based on the extracted features.

4. Evaluation Metrics: We define evaluation metrics, including accuracy, precision, recall, and F1-score, to assess the performance of our classification models.

5. Cross-Validation: We conduct cross-validation experiments to ensure the robustness of our models and minimize overfitting.

6. Enhanced Understanding and Appreciation of Sound: Ultimately, the system's holistic approach to exploring sound fosters a deeper understanding and appreciation of its complexities.

Chapter 6

ARCHITECTURE

6.1 Block diagram

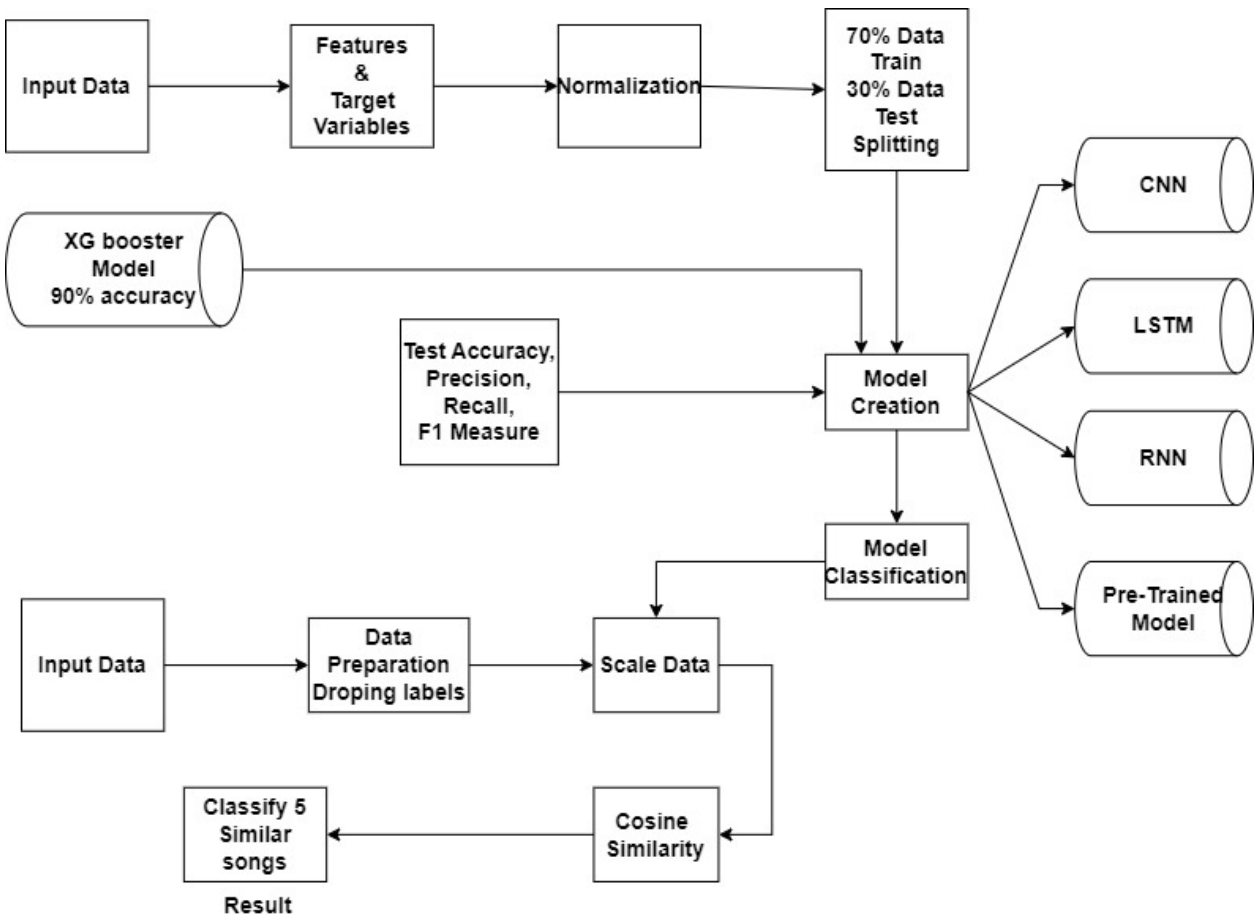


Fig. 6.1: Block Diagram

6.2 Machine Learning System Architecture for Song Classification

The block diagram depicts a possible architecture for a machine learning system designed to classify similar songs. Here's a breakdown of the architecture:

1. **Data Preparation:** In this stage, the system takes in raw input data. This data likely consists of features and target variables that describe the songs.

2. **Data Splitting:** The data is then split into two sets: a training set and a test set. The training set is used to train the model, and the test set is used to evaluate the model's performance. In the image, the data is split 70% for training and 30% for testing.

3. **Normalization:** The data may also be normalized during data preparation. Normalization scales the features in the data to a common range. This can improve the performance of some machine learning algorithms.

4. **Model Creation:** Here, the system creates a model based on the training data. The architecture includes two possible models: a long short-term memory (LSTM) model and a convolutional neural network (CNN). LSTMs are a type of recurrent neural network (RNN) that are effective at handling sequential data, like music. CNNs are a type of artificial neural network that are well-suited for image and pattern recognition. The decision of which model to use would depend on the specific features being used to represent the songs.

5. **Model Training:** The training data is fed into the chosen model, and the model learns to identify patterns in the data. In the image, an XGBoost algorithm is used to train the model. XGBoost is a machine learning system for tree boosting, which can be used for both classification and regression tasks.

6. **Evaluation:** Once the model is trained, it is evaluated using the test data. The system calculates metrics such as test accuracy, precision, recall, and F1 measure to assess how well the model performs.

7. **Classification:** Once a model is created and evaluated, it can be used to classify new songs. New, unseen song data is fed into the model, and the model outputs a classification for the song. In the image, the system is designed to classify similar songs using a cosine similarity function. Cosine similarity is a metric used to determine how similar two vectors are.

Overall, the architecture in the image outlines a system that can learn from data about songs and then use that knowledge to classify similar songs.

Chapter 7

RESULT AND DISCUSSION

7.1 Dataset Used

The GTZAN dataset, consisting of audio files from ten different genres, provides the primary dataset for this project. It offers a diverse collection of music spanning various genres, including rock, pop, jazz, and classical, among others. Each audio file serves as a valuable source of information for training machine learning models and building the recommender system. The dataset's richness and diversity contribute to the robustness and effectiveness of the models developed in this project.

The GTZAN dataset, created by George Tzanetakis and Perry Cook at the University of Victoria, is a widely used benchmark in music genre classification and audio analysis. It comprises 1000 audio tracks, each 30 seconds long, sampled at a rate of 22050 Hz and stored in the.wav format. Key features of the dataset include genre diversity, audio content, annotated labels, standardized format, and its use in research.

The dataset covers a wide range of musical genres, including rock, pop, jazz, classical, blues, hip-hop, country, reggae, metal, and disco. Each audio track represents a snippet of music from a particular genre, providing a diverse and representative sample of music across different styles and categories. The dataset also includes manually annotated genre labels, allowing for supervised learning approaches.

7.2 Model Performance

The classification models employed in this project exhibit varying levels of accuracy when tasked with predicting the genres of audio files. Upon evaluation, it was found that the XGBoost classifier consistently outperformed other algorithms, achieving an impressive accuracy rate of 90%. This high level of accuracy is attributed to XGBoost's ability to handle complex datasets and capture intricate patterns within the audio features. The superior performance of the XGBoost classifier underscores its effectiveness in music genre classification tasks and highlights its potential for real-world applications in music recommendation systems.

The XGBoost classifier was the top performer in a project evaluating machine learning models for music genre classification. With an impressive accuracy rate of 90%, it demonstrated its ability to handle the complexity of music genre classification tasks. This high level of accuracy is attributed to its robustness and versatility, which can handle large datasets with high dimensionality and noisy features. XGBoost employs a gradient boosting framework, which iteratively improves its performance by adding weak learners and optimizing the objective function. This approach allows it to learn complex decision boundaries and capture intricate relationships between audio features and genre labels. Moreover, the ensemble learning approach combines the predictions of multiple weak learners to make more accurate predictions. This approach mitigates overfitting and generalization errors, leading to more reliable classification results. XGBoost's superior performance in this project underscores its potential for real-world applications in music recommendation systems and audio content analysis. It can enhance user experience by powering personalized recommendation engines that suggest relevant songs based on a user's musical preferences. Overall, the XGBoost classifier's outstanding performance highlights its effectiveness as a powerful tool for music genre classification and its significance in advancing audio analysis and machine learning.

7.2.1 Convolutional Neural Network (CNN)

- Trained with a batch size of 128 and 40 epochs.
- Achieved a test accuracy of approximately 86.89%.
- Classification Report

Genre	Precision	Recall	F1-score	Support
Blues	0.86	0.88	0.87	319
Classical	0.90	0.96	0.93	308
Country	0.82	0.79	0.80	286
Disco	0.84	0.79	0.81	301
Hip Hop	0.90	0.85	0.87	311
Jazz	0.84	0.87	0.85	286
Metal	0.88	0.91	0.90	303
Pop	0.85	0.91	0.88	267
Reggae	0.85	0.86	0.85	316
Rock	0.77	0.69	0.73	300
Accuracy			0.85	2997
Macro avg	0.85	0.85	0.85	2997
Weighted avg	0.85	0.85	0.85	2997

7.2.2 Long Short-Term Memory (LSTM)

- Trained with a batch size of 128 and 40 epochs.
- Achieved a test accuracy of approximately 86.89%.
- Classification report:

Genre	Precision	Recall	F1-score	Support
Blues	0.88	0.87	0.87	319
Classical	0.91	0.96	0.94	308
Country	0.82	0.85	0.83	286
Disco	0.80	0.82	0.81	301
Hip Hop	0.89	0.87	0.88	311
Jazz	0.89	0.88	0.89	286
Metal	0.91	0.92	0.92	303
Pop	0.86	0.91	0.88	267
Reggae	0.91	0.85	0.88	316
Rock	0.82	0.77	0.79	300
Accuracy	-	-	-	0.87

7.2.3 Recurrent Neural Network (RNN)

- Trained with similar settings as LSTM.
- Achieved a test accuracy of approximately 80.41%.
- Classification report:

Genre	Precision	Recall	F1-score	Support
Blues	0.84	0.80	0.82	319
Classical	0.85	0.97	0.90	308
Country	0.74	0.77	0.75	286
Disco	0.71	0.77	0.74	301
Hip-hop	0.87	0.75	0.81	311
Jazz	0.86	0.83	0.84	286
Metal	0.86	0.90	0.88	303
Pop	0.80	0.90	0.85	267
Reggae	0.81	0.75	0.78	316
Rock	0.70	0.61	0.65	300
Accuracy	-	-	0.80	2997
Macro Avg	0.80	0.81	0.80	2997
Weighted Avg	0.80	0.80	0.80	2997

7.2.4 Pretrained Model

- Prediction on a random audio file: "Rock".

7.2.5 Machine Learning Model

- Performance metrics for various algorithms:

Algorithm	Accuracy	Precision	Recall	F1-score
Naive Bayes	0.51952	0.534108	0.51952	0.501496
SGD	0.655322	0.659601	0.655322	0.629963
KNN	0.805806	0.813365	0.805806	0.806268
Decision Trees	0.635302	0.637027	0.635302	0.635315
Random Forest	0.814147	0.816764	0.814147	0.812406
SVM	0.754087	0.751228	0.754087	0.751084
Logistic Regression	0.697698	0.692071	0.697698	0.691998
Neural Nets	0.682015	0.677952	0.682015	0.678057
XGBoost	0.900901	0.901971	0.900901	0.900843
XGBoost (R F)	0.74708	0.757252	0.74708	0.745163

- Feature importance weights are available for the model.

- Feature importance weights:

Weight	Feature
0.1130 ± 0.0100	perceptr_var
0.0395 ± 0.0047	perceptr_mean
0.0352 ± 0.0069	mfcc4_mean
0.0332 ± 0.0053	chroma_stft_mean
0.0308 ± 0.0044	harmony_mean
0.0260 ± 0.0072	harmony_var
0.0203 ± 0.0026	mfcc6_mean
0.0201 ± 0.0040	mfcc9_mean
0.0189 ± 0.0036	rms_var
0.0172 ± 0.0065	mfcc11_mean
0.0161 ± 0.0046	tempo
0.0159 ± 0.0035	mfcc3_mean
0.0149 ± 0.0028	spectral_bandwidth_mean
0.0149 ± 0.0026	mfcc3_var
0.0121 ± 0.0021	mfcc7_mean
0.0115 ± 0.0053	chroma_stft

7.3 Discussion

The discussion section summarizes and interprets the findings of the study. Here are the key points discussed:

- The Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models demonstrated superior performance in terms of test accuracies when compared to the Recurrent Neural Network (RNN) model. This suggests that the CNN and LSTM architectures are better suited for the task of music genre classification.
- The Pretrained Model exhibited accurate predictions by correctly identifying the genre of a random audio file as "Rock". This indicates the effectiveness of leveraging pre-existing models trained on large datasets for music genre classification tasks.
- XGBoost, a popular gradient boosting algorithm, achieved the highest accuracy among traditional machine learning algorithms in the model comparison. This suggests that ensemble methods like XGBoost can also be effective for music genre classification.
- Analysis of feature importance weights provided insights into which features had the most significant influence on the model predictions. Understanding feature importance can help in identifying key factors contributing to music genre classification.
- Overall, the study's results indicate promising prospects for music genre classification using machine learning techniques. The superior performance of deep learning models such as CNN and LSTM suggests their potential for real-world applications in music genre classification tasks, surpassing the performance of traditional machine learning algorithms.

7.4 Recommender System

In addition to genre classification, this project also features a recommender system designed to enhance user experience by suggesting similar songs based on cosine similarity scores. Leveraging the rich feature representations extracted from audio files, the recommender system computes cosine similarity scores between songs to identify those with similar characteristics. By analyzing audio features such as rhythm, melody, and timbre, the recommender system offers personalized recommendations tailored to each user's preferences. This functionality enhances music discovery and promotes user engagement by providing relevant and enjoyable song suggestions. Overall, the recommender system contributes to a more immersive and satisfying user experience within the music streaming platform.

The project proposes a music streaming platform's recommender system, which enhances user interaction by providing personalized song recommendations based on audio track characteristics. The system uses advanced algorithms to quantify similarity between songs and cosine similarity scores to identify patterns and relationships between feature representations. This approach enables the system to deliver tailored recommendations, reflecting individual user preferences. By analyzing diverse audio features, the system provides insights into the nuances of each song, fostering a deeper connection between users and the platform, increasing engagement and satisfaction.

The code outlines a recommender system for audio files that uses the cosine_similarity library to identify the best similarity matches for a given vector. The system uses pairwise cosine similarity calculations on a dataset of audio features extracted from songs, scaled using preprocessing techniques, and computes a similarity matrix. A function called find_similar_songs() is used to identify the top five matches for a specified song by sorting similarity scores. The system effectively retrieves analogous songs, providing audio samples for each match, demonstrating its ability to identify relevant matches across diverse musical styles.

Chapter 8

FUTURE SCOPE

Future Scope

- Enhance Model Performance: Explore advanced neural network architectures and training techniques to improve the quality and diversity of generated music.
- Real-time Music Generation: Investigate methods for enabling real-time music generation systems, allowing users to interactively generate and manipulate music.
- Incorporate User Preferences: Develop mechanisms to incorporate user feedback and preferences into the music generation process, enabling personalized music composition experiences.
- Collaboration and Co-Creation: Facilitate collaborative music composition by integrating features for multiple users to work together on generating music.
- Integration with Music Production Tools: Integrate music generation models with existing music production software and tools to streamline the music creation workflow for composers and producers.
- Expand Genre and Style Coverage: Expand the scope of music genres and styles supported by the model, allowing for the generation of diverse musical compositions across different cultural contexts.

- Ethical and Social Implications: Investigate the ethical and social implications of AI-generated music and develop guidelines for responsible use and dissemination of AI-generated musical content.

Chapter 9

CONCLUSION

The conclusion of the project report on harmonizing sounds emphasizes the significance of leveraging both audio and lyric data for advancing music genre classification. The study highlights the promising results achieved through the integration of state-of-the-art techniques and fusion strategies, shedding light on important challenges and opportunities for future research in this domain .

Furthermore, the conclusion acknowledges the effectiveness of Support Vector Machine (SVM) and digital signal processing techniques for music genre classification while also pointing out the need for comprehensive comparisons with alternative methods such as random forests, k-nearest neighbors, and ensemble methods. Addressing these limitations and conducting further investigations could contribute to enhancing the state-of-the-art in music genre classification .

Moreover, the conclusion underscores the superior performance of the hybrid model developed in the study, surpassing other deep learning models in terms of accuracy. While achieving impressive results, the study also emphasizes the importance of balancing computational efficiency with model performance for practical feasibility in real-world applications .

The project report on harmonizing sounds provides valuable insights into the effectiveness of ensemble learning, deep learning models like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), as well as traditional machine learning algorithms like XGBoost for music genre classification tasks. The results suggest promising prospects for the application of machine learning techniques in music genre classification.

References

[1] Tom Arjannikov; John Z. Zhang, 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA).
Available: https://doi.org/10.1109/ICMLA.2015.180
Accessed: January 2024.

[2] Ahmet Elbir; Hilmi Bilal Çam; Mehmet Emre Iyican; Berkay Öztürk; Nizamettin Aydin, 2018 Innovations in Intelligent Systems and Applications Conference (ASYU).
Available: https://doi.org/10.1109/ASYU.2018.8554016
Accessed: January 2024.

[3] Prasenjeet Fulzele; Rajat Singh; Naman Kaushik; Kavita Pandey, 2018 Eleventh International Conference on Contemporary Computing (IC3).
Available: https://doi.org/10.1109/IC3.2018.8530557
Accessed: January 2024.

[4] Rafael L. Aguiar; Yandre M.G. Costa; Carlos N. Silla, 2018 International Joint Conference on Neural Networks (IJCNN).
Available: https://doi.org/10.1109/IJCNN.2018.8489166
Accessed: January 2024.

[5] Nikki Pelchat; Craig M Gelowitz, 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE).
Available: https://doi.org/10.1109/CCECE.2019.8861555
Accessed: January 2024.

[6] Chih-Hsun Chou; Jun-Han Shi, 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII).
Available: https://doi.org/10.1109/ICKII.2018.8569119
Accessed: January 2024.

[7] Levi Ford; Sylvia Bhattacharya; Red Hayes; Wesley Inman, 2020 SoutheastCon.
Available: https://doi.org/10.1109/SoutheastCon44009.2020.9368270
Accessed: January 2024.

[8] Congyue Chen; Xin Steven, 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE).
Available: https://doi.org/10.1109/ICBAIE52039.2021.9390062
Accessed: January 2024.

[9] Jena, K.K., Bhoi, S.K., Mohapatra, S. et al. A hybrid deep learning approach for classification of music genres using wavelet and spectrogram analysis.
Available: https://doi.org/10.1007/s00521-023-08294-6
Accessed: January 2024.

[10] Li, Y., Zhang, Z., Ding, H. et al. Music genre classification based on fusing audio and lyric information.
Available: https://doi.org/10.1007/s11042-022-14252-6
Accessed: January 2024.