International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 9, Issue 8, August-2024, http://ijmec.com/, ISSN: 2456-4265

MUSIC GENRE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK

Syeda Zainab[1], Sumera Fatima [2], Syeda Saher[3], Dr.K.Nagi Reddy[4]

[1,2,3] B.E. Student, Department of IT, Lords Institute of Engineering and Technology, Hyderabad

[4] Professor, Department of IT, Lords Institute of Engineering and Technology, Hyderabad

k.nagireddy@lords.ac.in

Abstract

Feature extraction is essential in Music Information Retrieval (MIR), but traditional methods like MFCC are often ineffective for music genre classification. This study introduces an algorithm that utilizes spectrograms and Convolutional Neural Networks (CNNs) for better classification performance. Unlike MFCC, spectrograms capture detailed musical features like pitch and flux. Our approach uses a CNN to generate four feature maps from the spectrogram, which reflect trends over time and frequency. A subsampling layer reduces dimensions and improves resistance to pitch and tempo changes. The multi-layer perceptron (MLP) classifier achieves a 72.4% accuracy on the Tzanetakis dataset, surpassing MFCC's performance.

I. Introduction

Music genre classification has broad applications, ranging from organizing music libraries to developing recommendation systems. Since 2002, various manual feature extraction techniques have been proposed for this task. For instance, Tzanetakis [1] and Fu [2] reviewed and evaluated the effectiveness of low-level and mid-level acoustic features for genre classification. They found that relying on a single manually-selected feature often falls short of achieving high classification accuracy.

To address this, Bergstra [3] introduced aggregate features combined with an AdaBoost classifier to improve genre classification. Fu [4] also explored feature-level and decision-level combinations, demonstrating that combining features generally outperforms using a single feature.

Recently, deep learning methods have gained prominence in feature extraction for music genre classification. Hamel [5] proposed a feature extraction system utilizing Deep Belief Networks (DBNs) applied to Discrete Fourier Transforms

DOI: https://doi-ds.org/doilink/08.2024-93677473

(DFTs) of audio signals, using nonlinear SVM for classification.

Similarly, Andrew Y. Ng [6] employed shift-invariant sparse coding (SISC) to develop a high-level representation of audio inputs and also utilized Convolutional Deep Belief Networks (CDBNs) for audio classification tasks. In this paper, we propose a novel approach that uses Convolutional Neural Networks (CNNs) on spectrograms for music genre classification. Unlike traditional features such as MFCC, which often miss out on intricate musical details, spectrograms provide comprehensive information including pitch and flux.

Our approach begins by focusing solely on the amplitude component of the spectrogram, disregarding the phase information.

We then apply feature detectors (filters) to the spectrogram to generate four distinct feature maps that capture various musical components over time and frequency scales. Following this, we use a sub-sampling layer to reduce the dimensionality of the feature maps and improve the model's resistance to changes in pitch and tempo. The final high-level features are fed into a Multi-Layer Perceptron (MLP) classifier.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs), originally designed for digit recognition, are a biologically inspired variant of Multi-Layer Perceptrons (MLPs). CNNs integrate three core concepts to achieve shift, scale, and distortion invariance: local receptive fields, shared weights, and sub-sampling. These principles, adapted from image processing, are applied to spectrograms for music classification tasks.

Receptive Fields & Feature Detectors

The concept of receptive fields, first identified by Hubel and Wiesel in the visual system of cats, is used in CNNs to capture local features from an input image. We apply this idea to audio processing by treating the spectrogram as an image and using local feature detectors (filters) to extract various audio components.

These filters create feature maps that represent different aspects of the spectrogram, such as percussion, harmonic content, or pitch changes.

Sub-Sample Layer

Once feature maps are generated, a sub-sampling layer is applied to reduce their dimensions. This process involves selecting the maximum value from each sub-region in

DOI: https://doi-ds.org/doilink/08.2024-93677473

the feature map to decrease the overall data size and enhance invariance to pitch shifts and tempo variations. This layer helps manage the complexity of the model and improves its robustness against changes in musical input.

Finally, the processed feature maps are connected to an MLP classifier to perform genre classification. Our method demonstrates that spectrogram-based features, processed through CNNs, can offer significant advantages over traditional MFCC features for music genre classification.

II. Literature Survey

A. Overview of Music Genre Classification

Music genre classification is an essential task in music information retrieval (MIR) with applications in music recommendation systems, automatic playlist generation, and music organization. It involves categorizing music tracks into predefined genres based on their acoustic features. Traditional methods relied heavily on manual feature extraction, such as Mel-Frequency Cepstral Coefficients (MFCCs), which have limitations in capturing the full complexity of musical content [1]. Advances in computational methods and machine learning have shifted the focus towards automated and more effective approaches.

B. Machine Learning Approaches for Music Genre Classification

Several machine learning techniques have been employed for music genre classification, each with its strengths and limitations. Early approaches focused on manual feature extraction combined with traditional machine learning classifiers. Tzanetakis [2] and Fu [3] reviewed various low-level and mid-level features, such as spectral and temporal features, and evaluated their performance for genre classification. They found that while single features like MFCCs were useful, they often failed to achieve high classification accuracy on their own.

In response to these limitations, researchers began to explore feature aggregation methods. Bergstra [4] demonstrated that combining multiple features using ensemble methods like AdaBoost could significantly improve classification performance. Fu [5] expanded this approach by investigating feature-level and decision-level combinations, showing that combining different features could enhance genre classification results compared to using a single feature.

DOI: https://doi-ds.org/doilink/08.2024-93677473

With the advent of deep learning, more sophisticated methods emerged. Hamel [6] proposed using Deep Belief Networks (DBNs) on Discrete Fourier Transforms (DFTs) of audio signals for genre classification. This approach used a nonlinear Support Vector Machine (SVM) as the classifier, showing promising results. Andrew Y. Ng [7] introduced shift-invariant sparse coding (SISC) for learning high-level audio representations and employed Convolutional Deep Belief Networks (CDBNs) for audio classification, achieving better performance than traditional methods.

More recently, Convolutional Neural Networks (CNNs) have become a popular choice for music genre classification due to their ability to automatically learn hierarchical features from raw audio data. Qiuqiang Kong and Xiaohui Feng [8] proposed a CNN-based algorithm using spectrograms for genre classification. Unlike traditional features like MFCCs, spectrograms provide detailed information about both time and frequency components of music. Their approach involved using feature detectors to extract high-level features from spectrograms, which were then processed by a Multi-Layer Perceptron (MLP) classifier. This method achieved a classification accuracy of 72.4% on the Tzanetakis dataset, outperforming MFCC-based methods.

C. Analysis of Existing Research and Identified Gaps

The details summarizes key research contributions from 2002 to 2020, highlighting different algorithms and their effectiveness in music genre classification. While traditional methods like MFCCs and SVMs have been foundational, they have limitations in capturing complex musical patterns. Advances such as DBNs and CDBNs have improved performance but still face challenges in fully leveraging the intricate details of audio signals.

Recent developments in CNNs have demonstrated superior performance by utilizing spectrograms to capture both time and frequency information.

However, there is ongoing potential for refining these methods further, exploring new deep learning architectures, and integrating advanced feature extraction techniques.

Overall, while CNNs have shown significant promise, existing research indicates that there is still room for innovation. Future work could focus on enhancing CNN architectures, exploring hybrid models, and incorporating additional features to achieve

DOI: https://doi-ds.org/doilink/08.2024-93677473

even higher classification accuracy for music genre tasks.

III.System Analysis

Music genre classification systems were limited by their reliance on hand-crafted audio features and shallow machine learning models. These systems typically used manually engineered features such as Mel-Frequency Cepstral Coefficients (MFCCs), Constant-Q Transform (CQT), and Fast Fourier Transform (FFT), which were based on domain-specific knowledge rather than being tailored for the genre classification task. The machine learning techniques employed were primarily simple architectures, such as softmax regression and multi-layer perceptrons (MLPs). These basic linear and feedforward models were not sufficiently complex to capture the intricate relationships between raw audio data and musical genres. Additionally, these systems did not leverage advanced deep learning techniques, such as convolutional neural networks (CNNs), which are designed to learn feature representations directly from spectrograms. They also missed the opportunity to use transfer learning from pre-trained models on large datasets. Consequently, these early genre classification systems suffered from significant limitations due to their reliance on manually designed features and shallow learning models, which hindered their ability to achieve high classification performance.

The present system has several limitations. The key limitations characterize the traditional approaches used in music genre classification:

Hand-Crafted Features: Traditional methods depended on manually designed audio features like MFCCs, which may not fully capture the relevant information for genre classification. These features are based on human assumptions rather than learned directly from data.

Temporal Context: Features like MFCCs are extracted from short, isolated frames of audio, disregarding temporal patterns that can be important for genre classification.

Simple Linear Models: Models such as softmax regression have limited capacity for capturing complex patterns in audio features due to their linear nature.

Non-Linear Models: Although non-linear models like MLPs offer greater capacity for modeling complex patterns, their performance is still constrained by the quality of the input features.

Disjoint Training Processes: Traditional systems employed a pipeline approach that separates feature extraction, feature

DOI: https://doi-ds.org/doilink/08.2024-93677473

selection, and classifier training, rather than integrating these processes into an end-to-end learning framework.

Lack of Invariance: These systems were often sensitive to small variations in pitch or tempo, which could negatively impact classification accuracy.

Inability to Learn from Raw Data: Traditional methods relied on pre-engineered features rather than learning directly from raw audio signals or spectrograms.

Scalability Issues: Conventional methods struggled to scale with larger datasets compared to deep learning approaches that benefit from increased data availability.

In summary, the main disadvantages of these traditional systems are their dependence on hand-designed features, inability to incorporate temporal information, limited modeling capabilities, and lack of end-to-end learning processes. Deep learning techniques offer potential solutions to these issues by enabling more effective feature learning and classification.

Algorithms Used: The algorithms used in these existing systems included:

Hand-Crafted Feature Extraction: Techniques like MFCCs, chroma features, and spectral contrast were used as input features for machine learning classifiers such as Support Vector Machines (SVM), k-Nearest Neighbors (KNN), and Random Forests.

Feature Aggregation: Low-level features were aggregated using statistical methods like mean, variance, and histograms for classification purposes.

Automatic musical genre classification can help or even replace people in this process, making it a very useful addition to music information retrieval systems. Furthermore, automatic genre classification of music can provide a foundation for the generation and evaluation of features for any type of content-based musical signal analysis. Because of the rapid growth of the digital entertainment sector, the concept of automatic music genre classification has been highly popular in recent years.

Although categorizing music into genres is arbitrary, there are perceptual characteristics such as instrumentation, rhythmic structure, and texture that can help define a genre. Until now, digitally downloadable music had to be classified by hand. Automatic genre categorization algorithms would thus be a helpful addition to the development of audio information retrieval systems for music.

IV.Models and the System Study

DOI: https://doi-ds.org/doilink/08.2024-93677473

The feasibility study for the music genre classification project is an essential phase where the project's viability is assessed. This phase involves evaluating various aspects to ensure that the proposed system will be both beneficial and manageable for the organization. The feasibility study is broken down into three main considerations:

1. Economic Feasibility

Economic feasibility investigates the financial implications of developing the music genre classification system to determine if it is a cost-effective venture for the organization. This includes evaluating the costs associated with implementing machine learning algorithms for genre classification, which involves expenses for hardware, software, and development resources. Fortunately, many advanced machine learning libraries and frameworks are open-source, which helps keep the costs manageable. The primary expenses will include acquiring high-performance computing hardware, data collection for training models, and potentially some commercial software licenses for advanced tools. A thorough cost-benefit analysis indicates that the long-term benefits of a more accurate genre classification system—such as improved user experiences and enhanced music recommendation services—justify the initial investment. Thus, the project remains financially viable and aligns with budgetary constraints.

2. Technical Feasibility

Technical feasibility assesses whether the technical requirements of the proposed genre classification system can be met with the available resources. This involves evaluating the hardware and software needed for implementing machine learning models and processing audio data for genre classification. The technical demands are relatively straightforward, as contemporary hardware like an Intel Core i7 processor and 16GB of RAM is sufficient for running sophisticated machine learning models. Python, the selected programming language, offers a wide range of libraries for audio processing and classification tasks, including TensorFlow, Keras, and scikit-learn, all of which are well-documented and widely utilized. Consequently, the system's technical requirements are manageable, with no significant barriers to the implementation of the necessary technologies. This confirms that the system can be developed with the current technical resources.

3. Social Feasibility

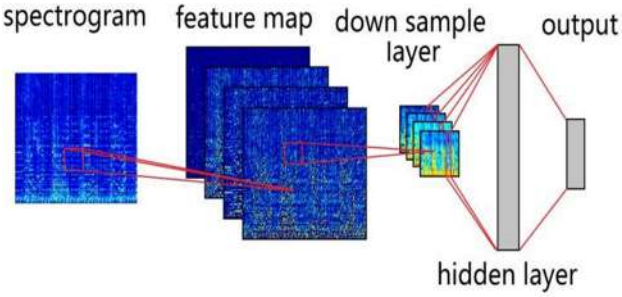
Social feasibility evaluates the potential acceptance of the music genre classification system by users and its impact on their

DOI: https://doi-ds.org/doilink/08.2024-93677473

interactions with music content. The system aims to offer an advanced tool for genre classification by analyzing audio features, which requires users to engage with the technology and understand its functionality. It is crucial to educate users about the advantages of this system and ensure they feel confident using it. This involves providing clear documentation, training materials, and ongoing support to help users understand how to utilize the system effectively. Additionally, the system must handle audio data responsibly and transparently, ensuring user trust and privacy. Positive user feedback and effective training will be key to the system's success and further refinements.

V. Frequency Domain Features

Logistic Regression (LR): For binary classification tasks, this linear classifier is commonly employed. The LR is implemented as a one-vs-rest approach for this multi-class classification assignment. That is, 8 binary classifiers are trained separately. During testing, the predicted class is picked from among the 8 classifiers with the highest probability.



Deep learning is a machine learning method that instructs computers to learn by doing what comes naturally to people. A computer model learns to carry out categorization tasks directly from images, text, or sound using deep learning. Modern precision can be attained by deep learning models, sometimes even outperforming human ability. A sizable collection of labelled data and multi-layered neural network architectures are used to train models.

Deep learning models are sometimes referred to as deep neural networks because the majority of deep learning techniques use neural network topologies.

The bubble chart is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

It is one of the most important modelling tools. It is used to model the system components. These components are the

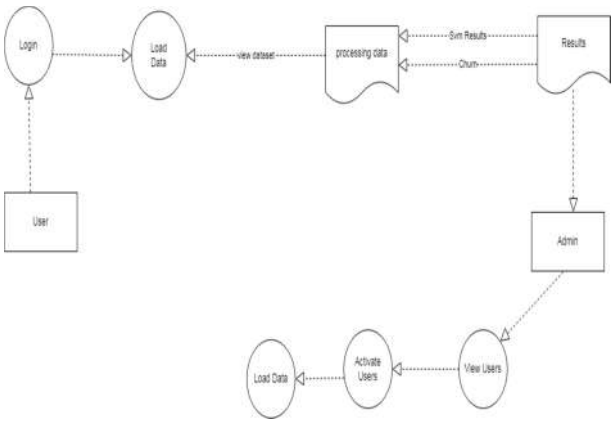
DOI: https://doi-ds.org/doilink/08.2024-93677473

system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

DFD shows how the information moves through the system and how it is modified by a series of transformations.

It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

It may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artefacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

The Primary goals in the design of the UML are as follows:

Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.

Provide extendibility and specialization mechanisms to extend the core concepts.

DOI: https://doi-ds.org/doilink/08.2024-93677473

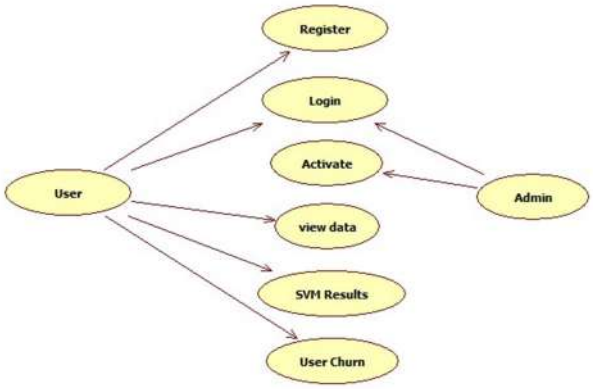
Be independent of particular programming languages and development process.

Provide a formal basis for understanding the modelling language.

Encourage the growth of OO tools market.

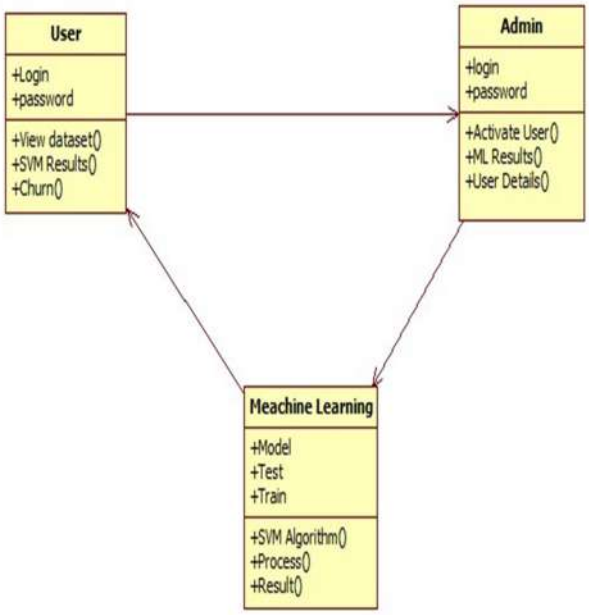
Support higher level development concepts such as collaborations, frameworks, patterns and components.

Integrate best practices.



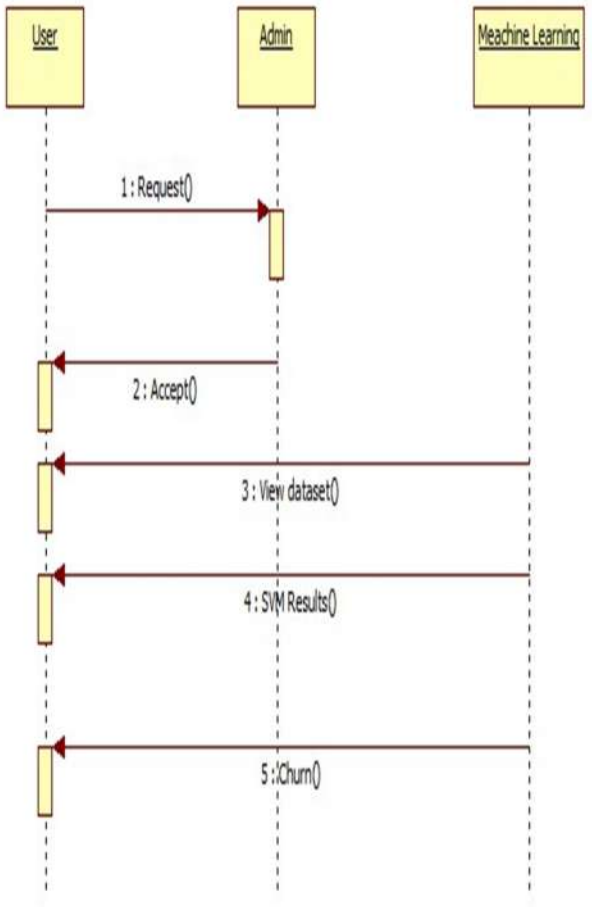
A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Class diagram:In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

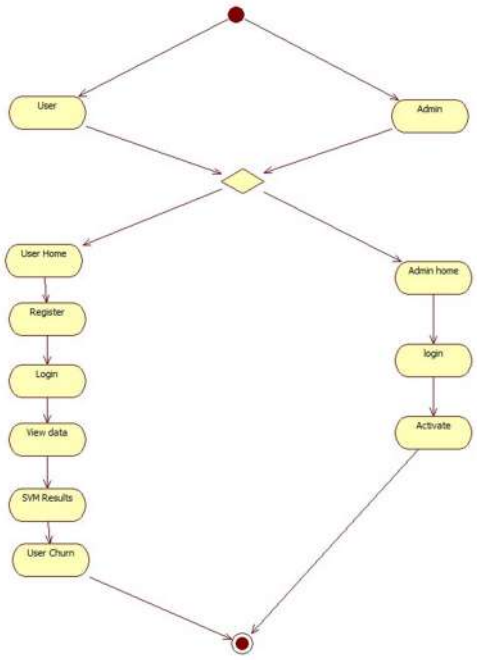


Sequence diagram: A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams

DOI: https://doi-ds.org/doilink/08.2024-93677473



Activity diagram: Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



VI.Modules Description

1. User:

The User starts by registering on the platform using a valid email address and mobile number for future correspondence. After registration, the Admin must approve the user account before the User can access the system. Once approved, the User can log in and upload audio files for music genre classification, ensuring that the audio files conform to the required format and structure for processing. The User can upload audio datasets with appropriate genre labels and also has the option to add new audio files to the existing dataset through the web interface provided by our Django application. On the web page, the User can

DOI: https://doi-ds.org/doilink/08.2024-93677473

select the "Classification" option to run genre classification algorithms, which will calculate and display metrics such as Accuracy, F1-Score, Precision, and Recall based on the selected models. The User can also select the "Prediction" option to upload a new audio clip and receive genre classification results, identifying the genre of the music as either rock, jazz, classical, or another genre.

2. Admin

The Admin logs into the system using their credentials to manage user accounts and oversee the operations of the genre classification system. Admin responsibilities include activating new user accounts to enable their access to the system. The Admin has full access to the system's data and can view and manage all datasets uploaded by Users. Admin can also navigate to the "Results" section of the web interface where the performance metrics of various genre classification algorithms are displayed, including Accuracy, F1-Score, Precision, and Recall. After the execution of all algorithms, the Admin can review the overall performance metrics on the web page to determine which algorithm achieved the best results for classifying music genres.

3. Data Preprocessing

Data preprocessing prepares raw audio files for machine learning tasks in music genre classification. This involves cleaning and transforming audio data for effective analysis. The preprocessing steps include noise reduction, handling missing data, and converting audio files into a suitable format for analysis. Techniques used in preprocessing include feature extraction methods like Spectrograms or MFCCs, normalization of audio features, and segmenting audio into frames for model training. The processed data is then structured for feature extraction and model development to ensure that the data is ready for effective genre classification.

4. Machine Learning

In this module, the preprocessed audio data is split into training and test datasets, typically using 80% of the data for training and 20% for testing. The cleaned and transformed audio data is then analyzed using various machine learning classifiers, such as Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Random Forests. ince the amount of data present is very less to effectively train a CNN we have created more data from this data, by dividing each song into 10 segments

DOI: https://doi-ds.org/doilink/08.2024-93677473

each having duration of 3 secs each. This allows us to increase the amount of data in order to train the CNN model more effectively The performance of these classifiers is evaluated based on metrics like Accuracy, Precision, Recall, and F1-Score. The results are displayed on the web page to facilitate comparison between algorithms. The classifier with the highest performance metrics is selected as the best model for music genre classification.

VII. Conclusion

Future research will focus on advancing convolutional neural networks (CNNs) by exploring methods to automatically learn feature detectors instead of relying on manual selection, which could lead to improved performance in music genre classification. Additionally, we plan to investigate the use of deeper CNN architectures with more layers to capture more abstract, high-level features, potentially enhancing the model's effectiveness in genre classification tasks.

In future work, we will aim to improve our convolutional neural network (CNN) models by shifting from manually selected feature detectors to techniques that allow for automatic feature learning, which could lead to better classification results. We will also explore adding more layers to the CNN to uncover more abstract features and further enhance the performance of genre classification.

Moving forward, our research will seek to enhance convolutional neural networks (CNNs) by developing methods for automatic feature detector learning rather than relying on manual techniques, which we believe could improve classification outcomes. Additionally, we will explore the potential benefits of deeper CNN architectures to extract more sophisticated, high-level features for better music genre classification.

Future work will involve advancing convolutional neural network (CNN) methodologies by exploring the automatic learning of feature detectors to replace current manual methods, which may result in better performance for genre classification. We will also investigate the effectiveness of increasing the number of layers in the CNN to achieve more abstract and higher-level feature extraction for improved classification accuracy.

VIII. References

[1] Tzanetakis G, Cook P. "Musical genre classification of audio signals". Speech and

DOI: https://doi-ds.org/doilink/08.2024-93677473

Audio Processing, IEEE transactions on, 2002, 10(5): 293-302.

[2] Fu Z, Lu G, Ting K M, et al. "A survey of audiobased music classification and annotation". Multimedia, IEEE Transactions on, 2011, 13(2): 303-319.

[3] Bergstra J, Casagrande N, Erhan D, et al. "Aggregate features and AdaBoost for music classification". Machine learning, 2006, 65(2-3): 473-484.

[4] Fu Z, Lu G, Ting K M, et al. "On feature combination for music classification". Structural, Syntactic, and Statistical Pattern Recognition. Springer Berlin Heidelberg, 2010: 453-462.

[5] Hamel P, Eck D. "Learning Features from Music Audio with Deep Belief Networks". ISMIR. 2010: 339344.

[6] Grosse R, Raina R, Kwong H, et al. "Shift-invariance sparse coding for audio classification". arXiv preprint arXiv:1206.5241, 2012.

[7] Elbir, Ahmet, and Nizamettin Aydin. "Music genre classification and music recommendation by using deep learning." Electronics Letters 56.12 (2020): 627-629.

[8] Jeong, Il-Young, and Kyogu Lee. "Learning Temporal Features Using a Deep Neural Network and its Application to Music Genre Classification." Ismir. 2016.

[9] Lau, Dhevan S., and Ritesh Ajoodha. "Music Genre Classification: A Comparative Study Between Deep Learning and Traditional Machine Learning Approaches." Proceedings of Sixth International Congress on Information and Communication Technology. Springer, Singapore,2022.

[10] Shaha, Manali, and Meenakshi Pawar. "Transfer learning for image classification." 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, 2018.

[11] Vishnupriya, S., and K. Meenakshi. "Automatic music genre classification using convolution neural network." 2018 International
Conference on Computer Communication and Informatics (ICCCI). IEEE, 2018.

[12] Silla, Carlos N., Alessandro L. Koerich, and Celso AA Kaestner. "A machine learning approach to automatic music genre classification." Journal of the Brazilian Computer Society 14.3 (2008): 7-18.

[13] MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam.

DOI: https://doi-ds.org/doilink/08.2024-93677473

[14] Chathuranga, Y. M. ., & Jayaratne, K. L. Automatic Music Genre Classification of Audio Signals with Machine Learning Approaches. GSTF International Journal of Computing, 3(2), 2013.

[15] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon, "An end-to-end neural network for polyphonic piano music transcription," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 5, pp. 927–939, 2016.

[16] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, Xavier Bresson. FMA: A Dataset For Music Analysis. Sound; Information Retrieval. arXiv:1612.01840v3, 2017.

[17] Keunwoo Choi, George Fazekas, and Mark Sandler, "Explaining deep convolutional neural networks on mu- sic classification," arXiv preprint arXiv:1607.02444, 2016.

[18] M. Goto and R. B. Dannenberg, "Music Interfaces Based on Automatic Music Signal Analysis: New Ways to Create and Listen to Music," in IEEE Signal Processing Magazine, vol. 36, no. 1, pp. 74-81, Jan. 2019, doi: 10.1109/MSP.2018.2874360.

[19] Y. M.G. Costa, L. S. Oliveira, C. N. Silla, "An evaluation of Convolutional Neural Networks for music classification using spectrograms", in Applied Soft Computing, Volume 52, 2017, Pages 28-38, ISSN 1568-4946, doi.org/10.1016/j.asoc.2016.12.024.

DOI: https://doi-ds.org/doilink/08.2024-93677473