JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

MUSIC GENRE CLASSIFICATION MACHINE LEARNING

[1]V. Neelima Devi, [2] Kakarla kiran,

[1]Head of Department, [2] MCA 2nd year,
[1]Master of Computer Application,
[1]Sanketika Vidya Parishad Engineering College, Visakhapatnam, India

ABSTRACT

Music genre is a classification system that classifies music into different types.The classification of music genre is very important to make a selection of songs from a large collection of data. Different features have been extracted as it is essential for the genre classification.We use convolution neural networks to classify music genres. The project aims to create an automated system for classification models of music genres.We use the most publicly used data set which is GTZAN for evaluation of music genres. It contains 10 genres(blues, classical, country, disco, hiphop, jazz, metal, pop,reggae, rock).The audio file is sent as input and the classification is done.CNN is used to classify the data furthermore.

KEYWORDS: Music genre Classification; Acoustic features; Machine Learning; Deep Learning.

INTRODUCTION

Music is so important to everyone's life, it brings out so many emotions in us like excitement. Music can change someone's mood, get them productive, the possibilities are endless.

Music Genre Recognition is an important field of research in Music Information Retrieval. A music genre is a conventional category that identifies some pieces of music as belonging to a shared tradition or set of conventions, i.e. it depicts the style of music.Music Genre Recognition involves making use of features such as spectrograms, MFCC's for predicting the genre of music.Here we are going to make use of GTZAN Dataset which is really famous in Music Information Retrieval (MIR). The Dataset comprises 10 genres namely Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae, Rock.Each genre comprises 100 audio files (.wav) of 30 seconds each.The model can learn the genre of the music by listening to the song 4-5 seconds without listening to the complete 30 sec song which takes more time to classify entire songs.We are going to use a Convolutional Neural Network, we need an image as an input, for this we will use the mel spectrograms of audio files and save the spectrograms as an image file.

BACKGROUND AND MOTIVATION

BACKGROUND

Genre classification is an important task with many real world applications. As the quantity of music being released on a daily basis continues to sky-rocket, especially on internet platforms such as Soundcloud and Spotify – a suggests that tens of thousands of songs were released every month on Spotify – the need for accurate meta-data required for database management and search/storage purposes climbs in proportion. Being able to instantly classify songs in any given playlist or library by genre is an important functionality for any music streaming/purchasing service, and the capacity for statistical analysis that correct and complete labeling of music and audio provides is in essentially limitless. Music genre classification is the task of automatically categorizing music into different genres, such as rock, pop, hip hop, jazz, and classical. This is typically done using machine learning algorithms that are trained on a large dataset of music with annotated genre labels. Features such as rhythm, melody, harmony, and timbre are extracted from the audio signal and used to make predictions about the genre of a piece of music.Evaluation of music genre classification models typically involves testing the model on a separate dataset of music that was not used for training. Performance is often evaluated using metrics such as accuracy, precision, recall, and F1 score.

MOTIVATION

The motivation for developing a music genre classification system using CNNs is to improve the accuracy and efficiency of music classification, which can have the various practical applications, including music recommendation systems, music streaming platforms, and the music search engines. Music is a universal language, and it has become an integral part of our lives. With the proliferation of music services and the sheer volume of available music, the need for automatic music genre classification has become more important than ever. The ability to automatically classify music by genre can enable personalized music recommendations, improve music search capabilities, and enhance the user experience on music platforms.Traditional methods for music genre classification, such as manual classification by music experts, can be time-consuming, expensive, and subjective. Furthermore, the human ear is limited in its ability to detect patterns and features in complex audio signals, especially in cases where the differences between genres may be subtle.

CNNs offer a promising solution to these challenges, as they can analyze large amounts of audio data and extract relevant features in an automated and objective manner. With large amounts of training data, CNNs can learn to recognize patterns and features in the audio signal that are indicative of a particular genre, and they can do so with high accuracy and speed.Furthermore, music genre classification using CNNs can enable personalized music recommendations that better match a user's preferences. By accurately classifying music into different genres, music streaming platforms can recommend music that is more likely to appeal to the user's taste, enhancing the overall user experience. Overall, the development of a music genre classification system using CNNs has the potential to enhance music platforms and services, improve music search capabilities, and enable more personalized music recommendations.

PROBLEM STATEMENT

Music plays a very important role in people's lives. Music brings like-minded people together and is the glue that holds communities together. Communities can be recognized by the type of songs that they compose, or even listen to. Different communities and groups listen to different kinds of music. This work exactly fulfills the above-mentioned requirement. This work will create a machine learning model that can predict the genre of the provided

REQUIREMENTS ELICITATION AND ANALYSIS

EXISTING SYSTEM

Most of the existing music genre recognition algorithms are based on manual feature extraction techniques. These extracted features are used to develop a classifier model to identify the genre. However, in many cases, it has been observed that a set of features giving excellent accuracy fails to explain the underlying typical characteristics of music genres. It has also been observed that some of the features provide a satisfactory level of performance on a particular dataset but fail to provide similar performance on other datasets. Hence, each dataset mostly requires manual selection of appropriate acoustic features to achieve an adequate level of performance on it.

PROPOSED SYSTEM

A proposed system for music genre classification using Convolutional Neural Networks (CNNs) could be as follows:

DATA PREPARATION

Collect a large dataset of music files from different genres, such as classical, jazz, rock, pop, etc. Split the data into training, validation, and test sets.

Feature Extraction: Extract features from the audio files that can be used as input to the CNN model. Some common features used for music genre classification include Mel-frequency cepstral coefficients (MFCCs), spectrograms, and chroma features.

MODEL ARCHITECTURE

Design a CNN model for music genre classification. The input to the model should be the extracted features from the audio files. The model should have multiple convolutional layers to capture local patterns in the input features, followed by pooling layers to reduce the spatial dimensions of the output. Finally, the model should have one or more fully connected layers to classify the input into different music genres.

TRAINING

Train the CNN model on the training set using a suitable loss function and optimization algorithm. Monitor the performance of the model on the validation set and adjust the model architecture and hyperparameters as necessary.

TESTING

Evaluate the performance of the trained model on the test set. Calculate metrics such as accuracy, precision, recall, and F1 score to measure the performance of the model.

Overall, a CNN-based system for music genre classification can achieve high accuracy and is robust to variations in the input features. However, it requires a large amount of labeled data and careful design and tuning of the model architecture and hyperparameters.

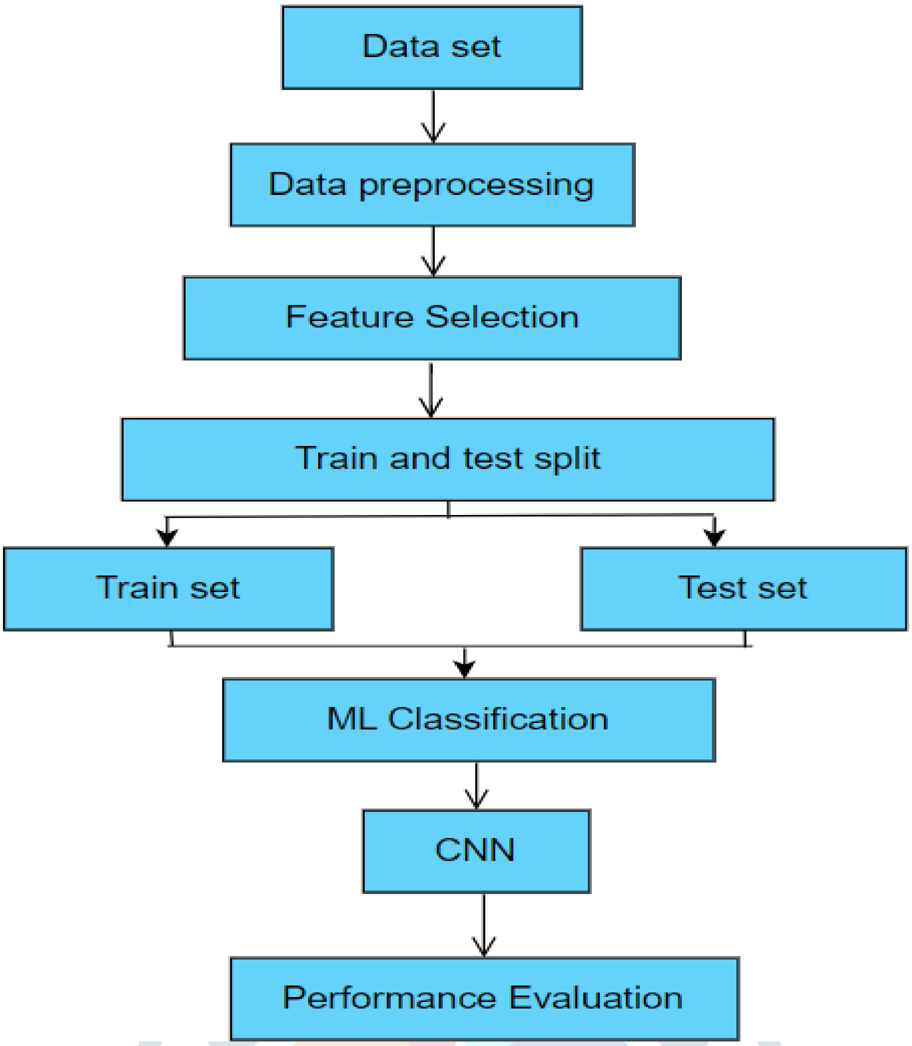


Fig: block diagram

Fig: music geners

Feasibility Study

A feasibility study on a music genre classification system would assess various aspects of the project to determine if it is viable and likely to succeed. Some of the factors to consider in such a feasibility study are:

Technical feasibility:

This involves evaluating the technical requirements of the system, including the hardware, software, and algorithms required to build a music genre classification system. This will also involve reviewing existing solutions and technologies for music genre classification to determine which would be best suited for the project.

Data feasibility:

This involves assessing the availability and quality of music data for training and testing the system. This includes the quantity of data, the variety of genres represented, and the quality of the annotations for genre labels.

Market feasibility:

This involves researching the demand for music genre classification systems and evaluating the competition in this market. This will involve reviewing existing products and services, as well as evaluating the target audience for the system.

Operational feasibility:

This involves assessing the processes and resources required to develop, deploy, and maintain the music genre classification system. This will include evaluating the skills and expertise of the team, aswell as the potential for scalability and growth.

SYSTEM DESIGN

WHAT IS SOFTWARE

Software, generally sense, is understood as a group of instructions or programs that instructs a computer to perform specific tasks. Software could be a general term that would describe computer programs. Scripts, applications, programs and a group of instructions are all different terms that would describe software.

The theory of software was first proposed by Alan Mathison Turing in 1935 in his essay "Computable numbers with an application to the Entscheidungsproblem." However, the word software was proposed by statistician and mathematician John Tukey in a very 1958 issue of yank Mathematical Monthly within which he discussed the electronic calculators' programs.

Software is typically divided into three categories

System software could be a base for application software. System software generally includes operating systems, device drivers, text editors, compilers, disk formatters and utilities helping the pc to regulate more efficiently. It's responsible for providing basic non-specific-task functionalities and management of hardware components. The system software is typically written within the language of C programming.

Programming software could also be a group of tools to help developers in writing programs. The numerous tools available are linkers, compilers, interpreters, debuggers and text editors.

Application software is typically used to perform certain tasks and also the samples of the applying software includes educational software, database management systems, office suites, application on gaming. the applying software can either be one program or a gaggle of portable programs.

SOFTWARE SPECIFICATION

Python Python is a high-level programming language which includes simplicity, consistency, access to great libraries and flexibility. Using this we can implement these ML algorithms with ease.

Python Libraries used

Numpy is an open-source library used in Python for array manipulations and appearing high stage operations in arrays of multi-size.

Pandas is an open-source python library used for data evaluation and manipulation which is read within the shape of a data frame.

MatplotLib is used for mathematically visualizing all our attributes and gives us various visualizations like plotting bar graphs, pie charts etc.

Sklearn is a ML library which includes diverse algorithms in- built like Linear regression, SVM, Random Forests and so on.

Training and Testing the Data

Train – Test Split: We have used 80% of the data for training and the other 20% of the data for testing. For that, We have used a library called “train-test split” for splitting the data set.

Performance Measures

Based on the performance measures like Accuracy, Precision, Recall and F1-Score, we find out which is the efficient model to classify the audio file into corresponding genre.

UML Diagrams

UML (Unified Modeling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It was initially started to capture the behavior of complex software and non-software system and now it has become an OMG standard. This tutorial gives a complete understanding on UML.

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.
UML stands for Unified Modeling Language.
UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints.
Although UML is generally used to model software system, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc.
UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object- oriented analysis and design. After some standardization.

Components of UML

UML diagrams are the ultimate output of the entire discussion. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete.
UML includes the following nine diagrams, the details of which are described in the subsequent chapters.
Class diagram Object diagram Use case diagram Sequence diagram
Collaboration diagram Activity diagram

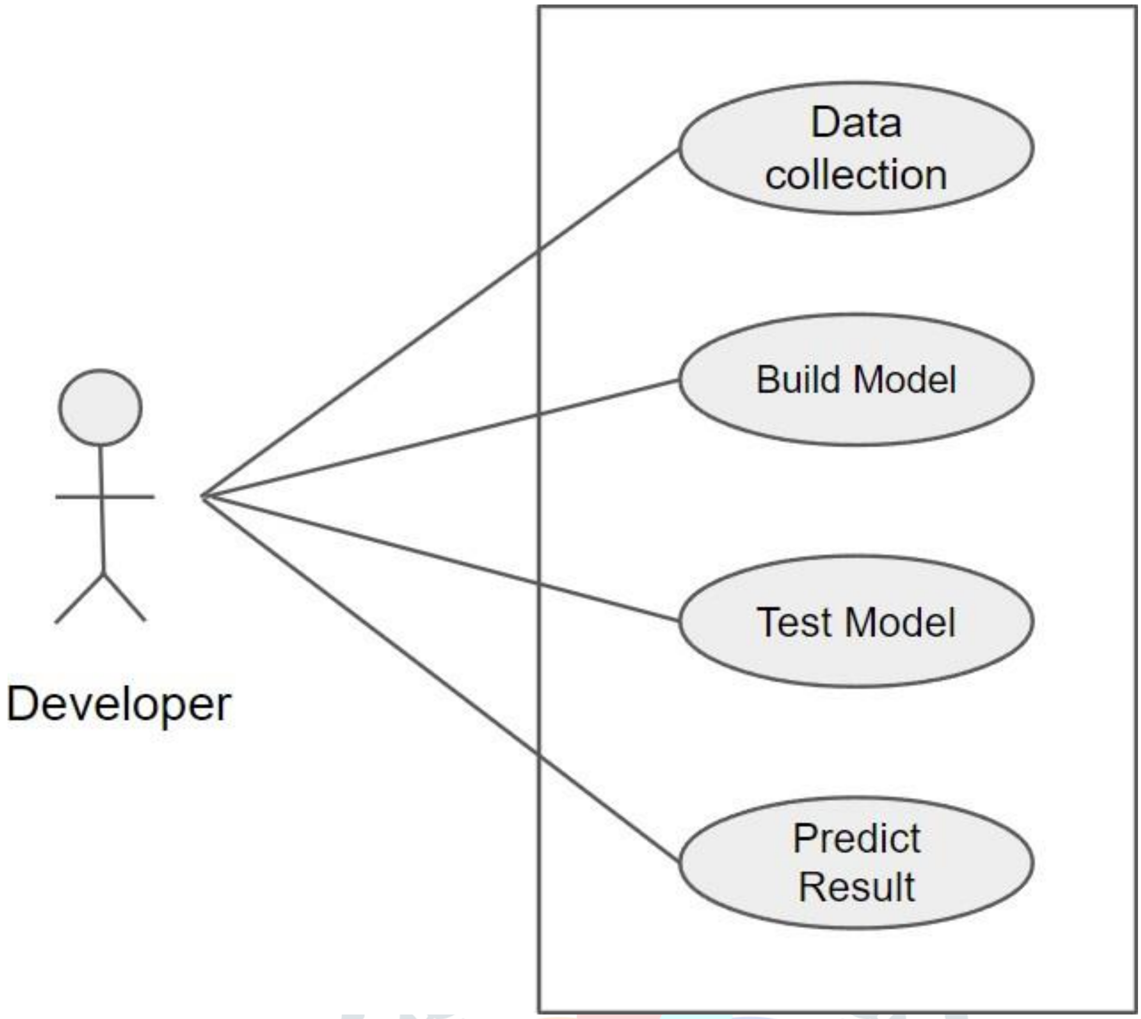
State chart diagram Deployment diagram Component diagram

The following are the main components of uml: - Use-case Diagram
Class Diagram Activity Diagram Sequence Diagram

Scenarios:
Scenario is a description of a specified sequence of actions. It depicts the behavior of objects undergoing a specific action series. The primary scenarios depict the essential sequences and the secondary scenarios depict the alternative sequences. A scenario is an instance of a use case; that is, A use case specifies all possible scenarios for a given piece of functionality. A use case is initiated by an actor. After its initiation a use case may interact with other actors, as well. A use case represents a complete flow of events through the system in the sense that it describes a series of related interactions that result from its initiations.

Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral 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 factors, their goals (represented as use-cases), and any dependencies between those use-case.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.



IMPLEMENTATION DESIGN

Google Colab is a web-based notebook environment that allows users to write, run and share code in a collaborative online setting. It is a free cloud-based service offered by Google, based on Jupyter Notebooks. The platform provides users with a Python environment and access to free resources such as GPUs and TPUs. Google Colab allows you to create and edit Jupyter notebooks in your browser without having to install anything on your computer. This makes it an ideal choice forpeople who want to work on data science and machine learning projects, but may not have the necessary hardware or software. With Colab, you can easily import datasets, write code, and collaborate with others on projects. You can also use Colab to store and share your notebooks with others through Google Drive or GitHub.One of the standout features of Colab is its ability to runcode on Google's powerful servers, which makes it possible to train large machine learning models using powerful GPUs and TPUs. Additionally, Colab provides built-in integration with many popular machine learning frameworks such as TensorFlow, PyTorch, and Keras.

Overall, Google Colab is a powerful and accessible tool for data science and machinelearning projects, which can be used by beginners and experts alike. Its ease of use and ability to collaborate with others make it an ideal choice for anyone looking to work on data science projectsin a collaborative and accessible way.Google Colab is a powerful tool for data science and machine learning projects. Here are some of the key features of Google Colab:

Free to Use: Google Colab is completely free to use and offers a variety of resources like GPU and TPU for accelerating your computations.

Collaborative Environment: Google Colab provides a collaborative environment, which allows multiple people to work together on the same notebook in real-time.

Jupyter Notebook Integration: Google Colab is based on Jupyter Notebook, which means that you can easily create, edit and share notebooks with others.

GPU and TPU Support: Google Colab offers free access to GPUs and TPUs, which can significantly speed up training time for deep learning models.

Scenario is a description of a specified sequence of actions. It depicts the behavior of objects undergoing a specific action series. The primary scenarios depict the essential sequences and the secondary scenarios depict the alternative sequences. A scenario is an instance of a use case; that is, A use case specifies all possible scenarios for a given piece of functionality. A use case is initiated by an actor. After its initiation a use case may interact with other actors, as well. A use case represents a complete flow of events through the system in the sense that it describes a series of related interactions that result from its initiations.

Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral 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 factors, their goals (represented as use-cases), and any dependencies between those use-case.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.

three or four convolutional layers it's possible to recognize handwritten numerals and with 25 layers it's likely to distinguish human faces. The program for this sphere is to stimulate machines to view the planet by way humans prepare, observe it in an extremelysimilar fashion and smoothly use the information for a mess of duty like image and video recognition, image examination and organization, media restoration, reference schemes, tongueprocessing, etc.

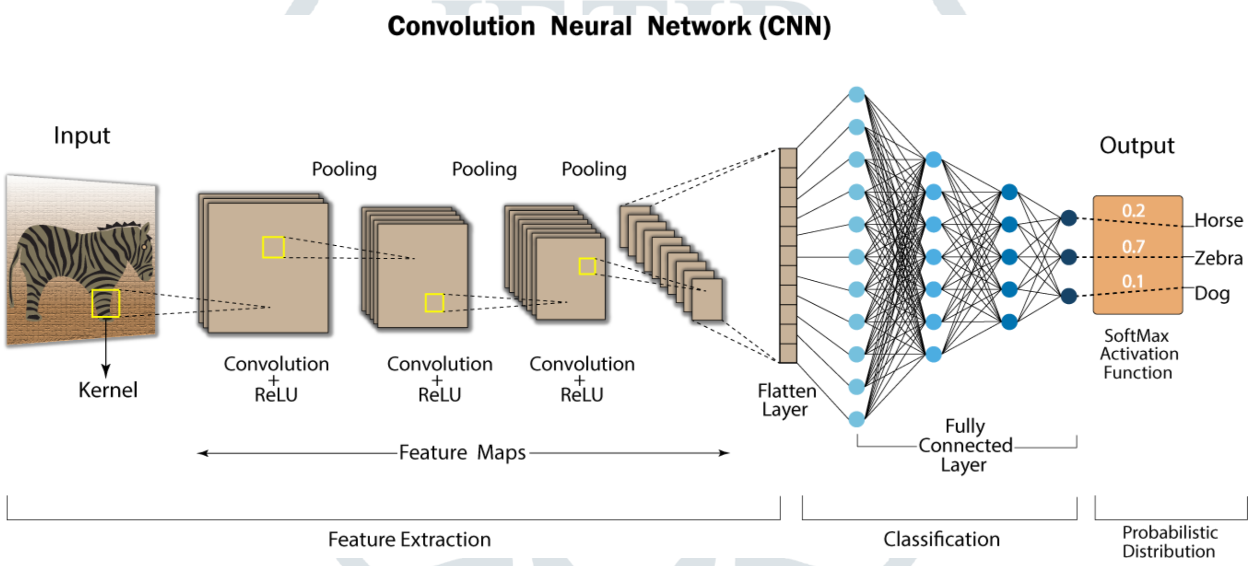


Fig-:CNN layers

Convolutional Neural Network Design: The structure of a convolutional neural network might be amulti-layered feed-forward neural network, created by constructing many unnoticed layers on cover of every other in a very specific method. It is the subsequent design that gives approval to CNN to betold categorized characteristics. In CNN, a number of them go along by sorting layers and unseen layers are normally convolutional layers followed by stimulation layers. The pre-processing taken in an exceedingly ConvNet is associated with the linked shape of neurons within the human brain and was inspired by the establishment of the visual area.

Data collection: Collect a large dataset of audio files consisting of multiple genres. This dataset should include a wide range of musical styles.

Feature extraction: Extract audio features that are relevant for the task of music genre classification. Popular features include Mel-Frequency Cepstral Coefficients (MFCCs), Chromafeatures, Spectral features, and Zero Crossing Rate (ZCR).

Data preprocessing: Preprocess the audio data to ensure that it is in a format that can be used by themodel. This may include resampling the audio to a common sample rate, normalization, and data augmentation to increase the amount of training data.

Model selection: Choose a suitable model for music genre classification, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or Support Vector Machines (SVMs). The choice of model depends on the size of the dataset, the complexity of the classification task, and the available computing resources.

Model training: Train the selected model using the preprocessed audio data. This involves feeding the model with the audio features and their corresponding labels and updating the model parameters to minimize the classification error.

Model evaluation: Evaluate the performance of the trained model on a separate test set using performance metrics such as accuracy, precision, recall, and F1-score.

Model optimization: Optimize the hyperparameters of the model to improve its performance. This may include tuning the learning rate, batch size, number of layers, and regularization techniques.

Deployment: Once the model has been trained and optimized, it can be deployed in a production environment for music genre classification. This may involve integrating the model into an application or web service that can accept audio files and return their predicted genre labels.

Overall, implementing a music genre classification system involves collecting and preprocessing audio data, selecting and training a suitable model, evaluating its performance, and optimizing its hyperparameters

ACCEPTANCE TESTING:

User Acceptance Testing is a critical phase of any project and requires significant

participation by the end user. It also ensures that the system meets the functional requirements.

Test Cases

SL.No	Test Case	Expected Result	Test Result	RS Id
1	Input file test case	yes	Successful	RS1
2	Genre selection test case	yes	successful	RS1
3	Feature extraction test case	yes	Successful	RS2
4	Large file handling test case	yes	Successful	RS3
5	Error handling test case	yes	Successful	RS4

CONCLUSION

In conclusion, music genre classification using Convolutional Neural Networks (CNNs) has proven to be a promising approach for automatic music genre classification. CNNs are able to learn effectivefeatures directly from the raw audio signals and can capture characteristics of music signals, which are essential for music genre classification.Various studies have shown that CNN-based models can achieve high accuracy rates for music genre classification, surpassing traditional feature-based models. However, the performance of CNNs depends on various factors such as the size of the dataset, the quality of the audio data, the architecture of the CNN model, and the choice of hyperparameters.Overall, music genre classification using CNNs is a promising field of research thathas the potential to revolutionize the way music is organized and curated in various applications suchas music recommendation systems, music libraries, and music streaming services.

REFRENCES

[1] A. Tzanetakis, G. and Cook, P. “Musical genre classification of audio signal”, IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 3, pp. 293-302, July 2002.

[2] Holzapfel, A. and Stylianou Y. “Musical genre classification using nonnegative matrix factorization-based features”, IEEE Transactions on Audio, Speech, and Language Processing, Vol. 16, No. 2, pp. 424-434, 2008.

[3] Benetos, E. and Kotropoulos C. “A tensor-based approach forautomatic music genre classification”, Proceedings of the European Signal Processing Conference, Lausanne, Switzerland, 2008

[4] Lidy, T., Rauber, A., Pertusa, A. and Inesta, J. “Combining audio and symbolic descriptors for music classification from audio”, Music Information Retrieval Information Exchange (MIREX), 2007.

[5] Bergstra, J., Casagrande, N., Erhan, D., Eck, D. and Kegl B. “Aggregate features and AdaBoost for music classification”, Machine Learning, Vol. 65, No. 2-3, pp. 473-484, 2006.

[6] A. Karatana and O. Yildiz, “Music genre classification with machine learning techniques,” in Signal Processing and Communications Applications Conference (SIU), 2017 25th, IEEE, 2017, pp. 1–4.

[7] Han, J., Kamber, M., & Pei, J. (2011). Data mining: concepts and techniques: concepts and techniques. Elsevier.

[8] Witten, I. H., & Frank, E. (2005). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

[9] Cortez P., In press. RMiner: Data Mining with Neural Networks and Support Vector Machines using R. In R. Rajesh (Ed.), Introduction to Advanced Scientific Softwares and Toolboxes.

[10] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5– 32, Oct. 2001.

[11] G. Gwardys and D. Grzywczak, “Deep image features in music information retrieval,” Intl Journal of Electronics and Telecomunications, vol. 60, no. 4, pp. 321–361, 2014.

[12]https://librosa.github.io/librosa/(visitedon15/06/2018)

[13] https://keras.io (visited on 15/06/2018)

[14] S. Dieleman. (2014). Recommending music on spotify with deep learning,[Online].Available: http://benanne.github.io/2014/08/05/spotifycnns.html (visited on 03/31/2018

[15] Ahmet Elbir, Hamza Osman Ilhan, Gorkem Serbes, Nizamettin Aydin. "Short Time Fourier Transform based music Genre classification" , 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT), 2018

BIBILIOGRAPHY

V.Neelima Devi Project Guide working as head of department in Master of computer application (MCA) Sanketika Vidya Parishad Engineering college, Visakhapatnam, Andhrapradesh. with 6 years of experience in master of computer applications(MCA),Accredited by NAAC. with her area of interests in C, Computer Organization, Software Engineering IOT.

Kakarla Kiran is studying his 2nd year, Master of Computer Applications in Sanketika Vidya Parishad Engineering College, affiliated to Andhra University, accredited by NAAC. With her interest in python, machine learning and as a part of academic project, she choose Image processing using python. As a result of a desire to comprehend the flaws in conventional reporting and to preserve timely and high-quality report output in Face mask recognition. A completely developed project along with code has been submitted for Andhra University as an Academic Project. In completion of her MCA.