**Abstract**

**Intelligent infotainment processes and music streaming services use automatic music genre detection as a vital tool for music retrieval, suggestion, and personalization. These systems can be particularly useful for in-car audio, as driver interaction with infotainment systems can be a major source of distraction. There are two key tasks to consider for improved genre classification: selecting an appropriate classifier and extracting suitable audio features. In this study we proposed the deep neural network for the automatic classification of music genres. The proposed model used the tunned hyperparameters extracted by the GridSearchCV for the classification. By following the training of the model, the proposed model showed a 0.55% test accuracy score.**

**Introduction**

**The internet has become a hub for sharing creative work, such as music, among artists and art enthusiasts across the globe. While music existed before the internet, the web has made it possible for individuals to access and collect an exponentially larger amount of musical content through platforms like Spotify, iTunes, YouTube, and more. This abundance of music has presented two challenges: the need for the automatic organization of music collections, as it is no longer feasible for users to do so manually, and the need to recommend new songs to users based on their listening habits. To address these challenges, it is necessary to have the ability to group songs into semantic categories.**

**Music genres are categories that have developed over time due to a variety of cultural, artistic, and market influences, and are used to categorize similar compositions and organize music collections. However, the distinctions between genres are often blurry, making the task of music genre recognition (MGR) challenging. Despite some debate over its utility due to its ambiguous and culturally defined nature, MGR is widely used and understood by end-users who find it helpful to discuss musical categories.**

**The goal of this challenge, one of four Web Conference challenges, was to identify the musical genre of a song based solely on a recording of the song. The genres are broad categories, such as pop or rock, and each song belongs to only one genre. Other metadata, such as the song title or artist name, was not allowed to be used in the prediction. The data for this challenge came from the Free Music Archive (FMA) dataset, which includes a large collection of Creative Commons-licensed audio from various artists and albums organized into 161 genres. The dataset includes high-quality audio, pre-computed features, and metadata such as track and user information, tags, and artist biographies.**

**Methodology**

**Dataset**

**The FMA (dataset for music analysis) was downloaded from GitHub for the classification of music based on their Genres. The FMA is a music dataset based on the music tracks and the metadata about the tracks. Each track or music in the dataset was based on the 30s duration. The dataset (small version) was also based on the 8000 tracks which were categorized into 8 different Genres. The name of the genres and the number of samples in each Genre is presented in Table 1.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Genre Name** | **Samples** | **Genre Name** | **Samples** |
| **Rock** | **1000** | **Hip-Hop** | **1000** |
| **Pop** | **1000** | **Folk** | **1000** |
| **International** | **1000** | **Experimental** | **1000** |
| **Instrumental** | **1000** | **Electronic** | **1000** |

**Preprocessing and Feature Extraction**

**Preprocessing the dataset is a crucial step for the training of machine learning and deep learning models. Firstly, the audio tracks were loaded using the Librosa library of python. After analyzing the audio tracks, it was evaluated the dataset was already cleaned and processed. Each track of the dataset was 30s long and properly class-balanced dataset. Further, the librosa library was also used for feature extractions. All the loaded tracks were passed to the library and librosa extract the 140 features for each audio track. After extracting the features, the dataset had converted into the shape of 8000\*140.**

**Train Test Split**

The dataset must be divided into multiple subsets for training, testing, and validating the model after preprocessing and feature extraction. The scikit-learn library's train-test-split method was used to split the dataset into three parts. The built-in train-test-split function randomly selects samples from each class, with a specified ratio for one subset and the remaining samples for the other subset. The observations in the resulting subsets are not overridden. We used the reprocessed data to divide the FMA collections into three separate subsets. The processed features dataset was then divided into a training set (70%), a validation set (10%) and a testing set (20%). After splitting the dataset, the train, validation and test sets contain 5600, 800 and 1600 samples, respectively. The number of samples in the train and test sets is shown in Table 2.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Training Set | Testing Set | Validation Set |
| Number of Samples | 5600 | 1600 | 800 |
| Number of Features | 140 | 140 | 140 |

**Model Training**

**We employed two different deep-learning models for the classification of music genres. Firstly, the MLP (multi-layer perceptron) was used with cross-fold validation to classify the audio track. The five cross-fold value was used during the training of the model. Further, the GridSearchCV was implemented using the MLP model. GridSearchCV is a built-in function of scikit-learn library that receives the multiple values for model hyperparameters. Further, it employed all combinations of received hyperparameters on the model and return the optimal parameters. The main objective of the grid search cv is to extract the optimal features for the training of the model. The GridSearchCV was implemented with the MLP model in the proposed work. Numerous hyperparameters including the learning rate, alpha, activation function, early stopping and optimizers were passed to GridSearchCV. `**

Table 1: Hyperparameters values for GridSearchCV.

|  |  |
| --- | --- |
| **Hyperparameter** | **Values** |
| **Hidden Layer sizes** | **10, 50, 100** |
| **Alpha** | **0.001, 0.01, 0.1** |
| **Learning Rate** | **0.001, 0.01, 0.1** |
| **Activation Function** | **identity, logistic, tanh, ReLU** |
| **Optimizers** | **LBFGS, SGD, Adam** |
| **Early Stopping** | **True, False** |

**After extracting the optimal features, the deep neural network was trained for the classification of the audio tracks. The proposed DNN network was based on the 12 dense layers. The first layer of the proposed model was used as the input layer with the shape 140 equivalent to the feature dimension. The last layer of the proposed model was used as the output layer with the shape 8 equivalent to the number of music Genres. A dropout layer with a 0.2 dropout value was used just before the output layer. The purpose of the dropout layer was to reduce the effect of model overfitting. The internal 10 layers of the proposed DNN model were designed in a u-network shape with 512 neurons to 16 neurons. The complete architecture of the proposed model is presented in Figure 1.**

Table

Description automatically generated with medium confidence

Figure 1: Architecture of the proposed model.

**Evaluation**

Researchers used various evaluation metrics, such as accuracy, precision, recall, and f1-score, to assess the classification model. Recall, also known as sensitivity, is the proportion of actual positive instances that correctly predict positive outcomes out of all true positive situations. Precision, or confidence, on the other hand, is the proportion of predicted positive cases that are actually true positives. In other words, recall measures "how many samples of a particular class are found among all samples of that class," while precision measures "how many of those samples are correctly classified." The harmonic mean of recall and precision is known as the f1-score. The trained models were evaluated using the test set, which contained 1600 audio tracks, using the formula provided in Equations 1-4.

Eq. 1

Eq. 2

Eq. 3

Eq. 4

**Results and Discussion**

**Environmental Setup**

**The python environment was established to perform the experiment of the proposed study. A Conda virtual environment with python version 3.9.2 was used to install the required libraries. Further, the TensorFlow version=2.3.0 and the latest versions of Numpy, matplotlib, pandas, and scikit learn were used for the experiments of the proposed study. the setup was established on Ubuntu:18.0 operating system with 32GB RAM and an 11G GPU unit.**

**In the proposed study, the FMA dataset was used for the classification of the music based on their Genres. For that, the dataset was split into training, testing and validation sets. Training and validation sets were used for the training of the models while the testing set was for the evaluation of the model. Firstly, GridSearchCV was used on the MLP model with 5 cross-fold validation and different hyperparameters values. GridSearchCV trained the MLP model for each combination of hyperparameters values passed to the function. It returned the optimal values as follows: learning rate: 0.01, alpha: 0.001, optimizer: Adam, activation function: ReLU and early stopping: True. The extracting optimal values of hyperparameters were used to train the MLP and DNN models.**

**The MLP model with training and validation set was trained on the tunned hyperparameters for music classification. MLP model showed a 0.41% accuracy score on the music tracks of the test set. The confusion metric on the test set was plotted for a better understanding of the results. The comprehensive classification report of the MLP model is presented in Table 2 for evaluation.**

Table 2: Classification Report - MLP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification Report for MLP Model** | | | | |
|  | precision | recall | f1-score | support |
| Electronic | 0.45 | 0.49 | 0.47 | 100 |
| Experimental | 0.32 | 0.34 | 0.33 | 100 |
| Folk | 0.19 | 0.17 | 0.18 | 100 |
| Hip-Hop | 0.66 | 0.7 | 0.68 | 100 |
| Instrumental | 0.44 | 0.42 | 0.43 | 100 |
| International | 0.5 | 0.53 | 0.51 | 100 |
| Pop | 0.28 | 0.27 | 0.28 | 100 |
| Rock | 0.55 | 0.5 | 0.52 | 100 |
|  |  |  |  |  |
| Avg/total | 0.42 | 0.43 | 0.42 | 800 |

**Later, the designed DNN model was also trained with the music tracks of the training set and validation set. The proposed DNN model showed 0.70% validation accuracy during the training of the model. Lastly, the performance of the DNN model was also evaluated on the test samples and showed the 0.55% accuracy score. The comprehensive classification report of the trained DNN model is also presented in Table 3.**

Table 3: Classification Report - DNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification Report for DNN Model** | | | | |
|  | precision | recall | f1-score | support |
| Electronic | 0.38 | 0.45 | 0.41 | 100 |
| Experimental | 0.15 | 0.13 | 0.14 | 100 |
| Folk | 0.14 | 0.16 | 0.15 | 100 |
| Hip-Hop | 0.52 | 0.59 | 0.55 | 100 |
| Instrumental | 0.37 | 0.29 | 0.32 | 100 |
| International | 0.27 | 0.29 | 0.28 | 100 |
| Pop | 0.22 | 0.18 | 0.2 | 100 |
| Rock | 0.44 | 0.43 | 0.44 | 100 |
|  |  |  |  |  |
| Avg/total | 0.42 | 0.43 | 0.42 | 800 |

**Comparatively, the proposed DNN model showed a higher accuracy relative to the MLP model. Initially, the DNN model showed a 0.46% accuracy score without optimal hyperparameters values. But the evaluation of the DNN model on optimal parameters expressed data the GridSearchCV play a significant role in the classification of music tracks. Further, the DNN and MLP model trained were trained once again on the optimal parameters. The high accuracy score of the proposed model on optimal parameters relative to the MLP model also expressed that the proposed model played a role in the robustness of the classification model.**

**Conclusion**

**This challenge was part of a broader effort to promote open evaluation in machine learning for music data, with the release of the open FMA dataset as the first step [1]. The goal of this initiative is to establish a reference benchmark based on open data. MIR research has historically been hampered by the lack of publicly available benchmark datasets, which are often restricted by copyright due to commercial interests in music by record labels. In the proposed study, we proposed a deep neural network for the music tracks. The proposed model showed a 0.55% test accuracy. The automatic classification of music tracks will help in the music archival systems to manage the songs. In future studies, the accuracy of the music tracks classification can be increased by extracting the features with already trained deep learning model like VGG-16 and VGG-19. Moreover, the complexity of the DNN model can be increased for the proposer learning of the model that ultimately increase the accuracy of music tracks classification.**