Music Genres Classification From Audio Data

Naredla Vamshi
Computer Science and engineering
Lovely Professional University
Punjab, India
vamshi.naredla123@gmail.com

Boorla Sai Teja
Computer Science and engineering
Lovely Professional University
Punjab, India
saiteja2k2@gmail.com

Marri Srikanth
Computer Science and engineering
Lovely Professional University
Punjab, India
marrisrikanth546@gmail.com

P.Omkar
Computer Science and engineering
Lovely Professional University
Punjab, India
vakitiomkar4134@gmail.com

Anjana R
Upgrad Campus, Upgard Education Private Limited
Bangalore, Karnataka, India
v_anjana.r@upgrad.com
ORCID: 0009-0000-9299-0790

Abstract— **In order to categorize and comprehend music based on shared traits and customs, genres are essential. Accurately categorizing musical genres not only improves listening to music as a whole but also makes it easier to assess artistic creativity and Caliber. In this paper, we describe an improved approach for classifying music genres using an Artificial Neural Network (ANN) Model. To accomplish this, we collected a comprehensive and properly classified music dataset, such as the widely-used FILM MUSIC AND ASHAMALUEV Music Genre dataset. We created feature maps by extracting spectral representations from the music data, which we then used to train the ANN Model. The model demonstrated outstanding performance, with a training accuracy of 97% and a validation accuracy of 89%, demonstrating its ability to reliably classify music genres. Furthermore, we focused on minimizing validation losses to optimize the model's performance. The evaluation matrix was computed to provide a comprehensive assessment of the system's classification capabilities. Our goal was to make the trained model easier to use for categorization, therefore we deployed it on a server with the Flask Web Framework along with it. Our proposed system not only outperforms the existing system based on Artificial Neural Networks, but also offers a more advanced and accurate solution for music genre classification. Our system's outstanding training and validation accuracies demonstrate its potential to provide users with a dependable and fast tool for categorizing and exploring the vast variety of music genres.**

Keywords—Music, genre, classification, ANN, signal processing, REST API, Flask.

I. INTRODUCTION

Music is defined as the arrangement of sounds in certain patterns. Everyone on the planet appears to be intrigued and interested in listening to various genres of music, whether it be for cultural or personal reasons. A style that highlights, mutes, or eliminates common components of a structured sound, such rhythm, loudness, and pitch, is referred to as music.

The traditional concept of music genre is characterized as a crucial component of global tradition. The two types of music are very different from one another. Experts believe there are other ways to group music into genres; the most popular ones include pop, rock, jazz, metal, blues, disco, and so forth. Music enthusiasts use technology to distinguish between these genres and to listen to the music that they like. To go from jazz to rap, listeners just need one other listener, but many listeners are certainly required to inspire a love of music. It might be challenging to arrange music files in the Music Information Retrieval (MIR) system according to genre. For the purpose of obtaining music from vast libraries, automatic genre classification of music is essential. It finds real applications in various areas, such as automatic tagging of unknown music.

The idea of genre classification, which divides music into different categories, aids in differentiating between two genres according to their rhythms. Genre classification has gained a lot of popularity recently, and new genres are appearing all over the world. These days, various online music services categorize music according to its genres using various approaches. Among the popular platforms are Spotify, JioSaavn, and others. Artificial neural networks are a type of deep neural network utilized extensively in deep learning for the processing of visual images.

The primary goal of this research project is to use ANN to more accurately define music genres. In this work, we experimented with the sample size of the FILM MUSIC AND ASHAMALUEV Music dataset, which comprises 100 pieces of music per genre, each lasting 30 seconds. Compared to earlier research efforts, we are able to obtain substantially better categorization results by expanding the dataset.

II. LITERATURE REVIEW

Musical Genre Classification of Audio signals.
AUTHORS: Tzanetakis, George & Cook, Perry [1].
It provides a comprehensive overview of techniques and methodologies employed in the analysis and classification of music genres based on audio features. The book offers valuable insights into the challenges and advancements in this field, making it a seminal resource for researchers and practitioners interested in music information retrieval and signal processing.

Automatic Music Genre Classification using Convolution Neural Network --- AUTHORS: S. Vishnupriya and K. Meenakshi [2].
The book delves into the utilization of CNNs' powerful feature extraction capabilities for analyzing audio data, offering insights into the effectiveness of deep learning techniques in this domain.

Music Genre Classification and Feature Comparison using ML
AUTHORS: Zhengxin Qi, Mohamed Rahouti, Mohammed A. Jasim, and Nazli Siasi [3].
This book provides a comprehensive exploration of various ML approaches employed in classifying music genres, highlighting the effectiveness of different feature extraction methods.

Music Genre Classification for Indian Music Genres
AUTHORS: Kumaraswamy, Balachandra & Shukla, Tushar & Swati, & Satyam, Kumar [4].
This book offers valuable insights into the challenges and techniques involved in categorizing Indian music genres using machine learning algorithms. Additionally, the significance of feature extraction, dataset selection, and model evaluation methods specific to Indian music genres.

III. MUSIC GENRE CLASSIFICATION

A. Music Genre

One of the most crucial characteristics to recognize a piece of music is its genre [1]. A music's behaviour can be characterized by its genre. People are affected differently by different genres. The outcomes of the categorization procedure can support research into sociopsychological facets of how individuals identify commonalities in music and create musical communities. There are many ways to divide music into different genres. The most well-liked musical genres include pop, reggae, rock, hip hop, jazz, blues, classic, country, and disco. However, determining a piece of music's genre can be challenging. This is not a particularly simple task.

B. Artificial Neural Network

Artificial Neural Networks (ANNs) are a fundamental type of machine learning model inspired by the structure and functioning of the human brain.
ANNs are made up of interconnected nodes, or artificial neurons, which are grouped into layers. After receiving and processing input signals, each neuron produces an output signal. An input layer, one or more hidden layers, and an output layer are the standard layers.
ANNs are versatile and can be applied to various tasks, including classification, regression, clustering, and pattern recognition. They are capable of learning complex patterns and relationships from data without being explicitly programmed.
In order to reduce the discrepancy between the expected and actual output for a given input, training an artificial neural network (ANN) entail modifying the weights and biases of connections between neurons. This is often done using optimization algorithms such as gradient descent.
Although ANNs can be powerful, they may face challenges with large datasets or complex problems due to issues like overfitting or vanishing gradients. However, advancements such as regularization techniques, different activation functions, and deeper architectures have helped address some of these challenges.

C. Exploratory Data Analysis of Audio data:

Before beginning any preprocessing, we will learn how to load audio files and visualize them in the form of a waveform. You can use the IPython library and provide it the audio file path directly if you want to load and listen to the audio file. We have selected the first audio recording from the dog bark category located in the fold 1 folder.
We will now load audio data via Librosa. For this reason, Librosa provides us with two options when we load any audio file. A two-dimensional array is the other, and sample rate is the first. Let's open Librosa, import the audio file mentioned above, and use it to plot the waveform.

Sample rate: The number of samples recorded in a second is represented by this number. When reading a file, librosa has a default sample rate of 2,800. The sample rate varies depending on the library you select.
2-D Array: The recorded amplitude samples are shown on the first axis. Additionally, the quantity of channels is shown on the second axis. There are two sorts of channels: stereo (which has two channels) and monophonic (which has one channel).

Librosa loads the data, normalizes it all, and attempts to provide it at a single sample rate. The Scipy Python Library allows us to accomplish the same thing. It will also provide us with two other kinds of information: data and sample rate.

Scipy prints the sample rate differently than Librosa does. Let's now visualize the audio wave data. One key distinction between the two is that although the data obtained by Librosa can be normalized for printing, it cannot be normalized for scipy audio file reading. The following three factors explain why Librosa is becoming more and more popular for audio signal processing.

1. It tries to converge the signal into mono (one channel).
2. It can represent the audio signal between -1 to +1(in normalized form), so a regular pattern is observed.
3. It has sample rate capability as well; by default, it transforms it to 22 kHz, but for other libraries, we see it based on a different value.

D. Problem statement

As previously stated, categorizing music into genres is not an easy or straightforward operation. As a result, employing neural networks to automate the classification operation can be useful and efficient. One class of deep neural networks that is frequently employed for visual image analysis is Artificial neural networks. Here, we used ANN to tackle the claim that we could categorize a musical genre according to its spectrographic representation.

IV. Methodology

The suggested work's methodology comprises gathering accurately categorized music datasets according to genre, preprocessing the data and extracting feature vectors from it, creating and classifying a model, and then deploying the model. The work's flow diagram is displayed below (Fig 1)—

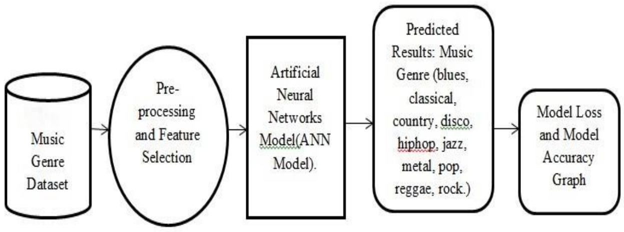

Fig.1

A. Music and Dataset

The dataset consists of 3000 audio recordings that are 10-second soundbites. Ten genres make up the collection, as indicated in table 1's first column below. Every one of these categories has one hundred (300) tracks. These are 16-bit, monaural, 22050 Hz audio files in the.wav format. The data is displayed as follows in Table I:

TABLE I. TABLE FOR DATA DISTRIBUTION

Genre	Music Count
Classical	300
Hip-hop	300
Disco	300
Country	300
reagge	300
jazz	300
Metal	300
Blues	300
Pop	300
Rock	300

B. Feature Vector Extraction

Certain audio recordings are made at a different frequency, such as 44 KHz or 22 KHz. The data will be at 22 KHz when using librosa, and we will then be able to view it in a normalized pattern. We now have to extract some significant information from the audio stream and save our data as independent (features extracted from the audio signal) and dependent (class labels) features. Mel Frequency Cepstral coefficients will be utilized to extract separate characteristics from audio streams.
MFCCs – Over the window size, the frequency distribution is summarized by the MFCC. Therefore, an analysis of the sound's frequency and temporal characteristics is conceivable. We will be able to recognize characteristics for categorization with this audio representation. As a result, it will attempt to transform audio into features that will aid in classification by using temporal and frequency characteristics. You can watch this video and read this Springer research paper to learn more about MFCC.
In order to show how MFCC is used in practice, let's start by applying it to a single audio file that we currently use.

We now need to construct the dataframe and extract features from each audio sample. Thus, we'll write a function that accepts the filename (or file path if it exists). Using librosa to load the file, we obtain two pieces of information. MFCC for the audio data will be found first, and the mean of the transpose of an array will be found to determine scaled features.
We now need to run a loop over every row in the dataframe in order to extract every characteristic for every audio file. To monitor the development, we additionally make use of the TQDM Python Library. For every file, we will set up a unique file path inside the loop and run the method to extract MFCC features, append features, and add labels in a newly formed dataframe.

C. Model Set up and Classification

Splitting the dataset

There will be training and testing sets within the dataset. Divide the dataset into test and train sets. There are 20% test and 80% train data. This will be carried out in order to test the model on untested data in order to assess its correctness, validate the model's performance, and train the model on a subset of the data. Divide the dataset into test and train sets. There are 20% test and 80% train data.

Audio Classification Model Creation**:** From the audio sample and splitter in the train and test sets, we have extracted characteristics. We shall now put an ANN model into practice. Our output shape, or the number of classes, is 10. We will build an artificial neural network (ANN) with three dense layers; the architecture is described below.

1. First layer: one hundred neurons. The number of features with an activation function of Relu indicates that the input shape is 40. To prevent overfitting, the Dropout layer will be used at a rate of 0.5.

2. Second layer: 200 neurons with activation function as Relu and the drop out at a rate of 0.5.
3. Third layer: 100 neurons with activation as Relu and the drop out at a rate of 0.5.

Compile the Model

In order to build the model, we must define the accuracy metrics (categorical cross-entropy), the loss function (categorical cross-entropy), and the optimizer (Adam).

Train the Model

The model will be trained and saved in.h5 format. A model with 450 epochs and a batch size of 30 will be trained. Callback, a checkpoint, will be used to determine how long it takes to train over the data.

D. Frameworks Utilized

Frontend Frameworks

HTML: Hypertext Markup Language (HTML) serves as the foundation for structuring the webpages of the classification system. HTML allows for the creation of a welcome page, login page, and file upload interface. The welcome page presents users with the system's title, providing a clear indication of its purpose and functionality.

CSS: Cascading Style Sheets (CSS) are utilized to enhance the visual presentation of the webpages. By applying CSS styles, the classification system achieves a professional and cohesive aesthetic across all elements, improving user experience and accessibility.

JavaScript: JavaScript enhances the interactivity of the classification system, enabling dynamic features such as form validation and asynchronous communication with the backend server. JavaScript facilitates the implementation of client-side functionalities, including user input validation on the login page and file upload preview before submission.

Backend Frameworks

Flask: Flask, a lightweight Python web framework, powers the backend of the classification system. Flask facilitates the development of a RESTful API to handle user authentication, file uploading, and model prediction requests. With Flask, the system maintains a secure and efficient communication channel between the frontend and backend components.

Integration of API

API (Application Programming Interface): The classification system incorporates an API to extend its functionality and enable interoperability with external systems and services. The API serves as an interface for accessing the system's features and performing operations such as user authentication, file uploading, and model prediction.

Authentication Endpoint: Handles user authentication requests, verifying user credentials and generating access tokens for authenticated users.

File Upload Endpoint: Accepts file upload requests from users, allowing them to submit audio files for genre classification.

Prediction Endpoint: Processes prediction requests, utilizing machine learning models to classify uploaded audio files into music genres and returning the classification results to the user.

RESTful Architecture: The API follows a Representational State Transfer (REST) architecture, adhering to RESTful principles such as statelessness, resource-based URL structure, and uniform interface. This architectural style ensures scalability, flexibility, and ease of integration with other systems and technologies.

System Workflow

The classification system follows a structured workflow to provide users with a seamless experience:

Welcome Page: Upon accessing the system, users are greeted with a welcome page displaying the system's title, creating an inviting entry point for exploration.

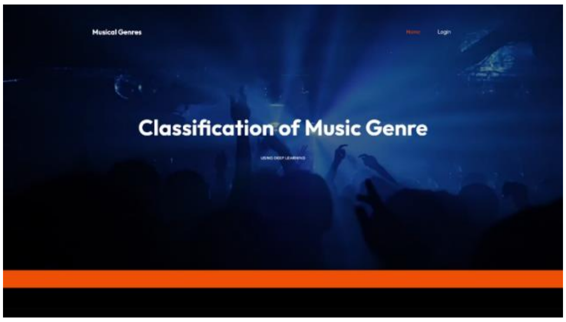

Fig.2 Welcome Page

User Authentication: Users authenticate themselves through the login page, where their credentials are validated by making requests to the authentication endpoint of the API. Upon successful authentication, users receive access tokens, which are used to authorize subsequent requests to protected endpoints.

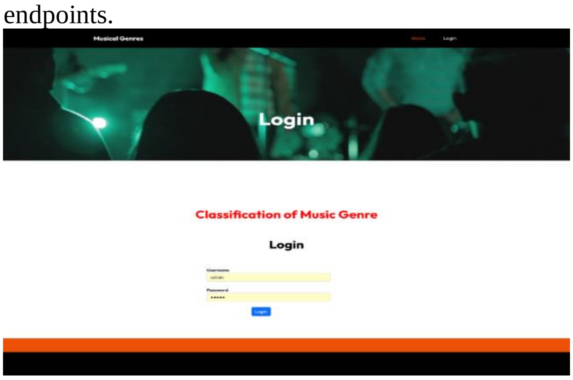

Fig.3 User Authentication

File Upload Interface: Users utilize the file upload interface to submit audio files for genre classification. When a file is uploaded, a request is sent to the file upload endpoint of the API, which processes the uploaded file and stores it in a secure location for further processing.

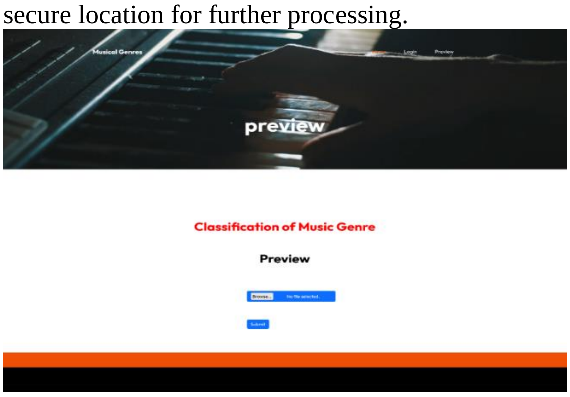

Fig.4 File Uploading Interface

Prediction: After file submission, users trigger a prediction request by interacting with the system's interface. The frontend communicates with the prediction endpoint of the API, which invokes machine learning models to classify the uploaded audio file into music genres. The classification results are returned to the user through the frontend interface, providing valuable insights into the predicted genres.

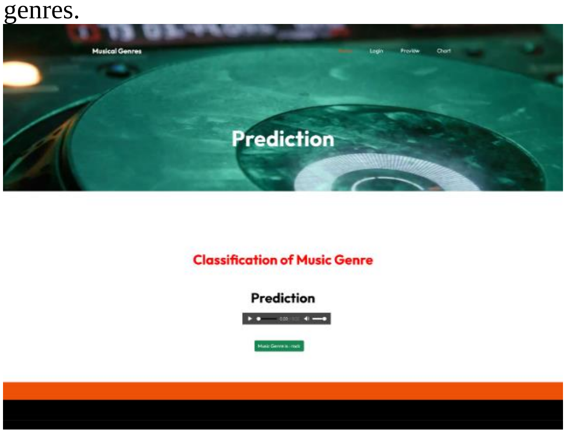

Fig.5 Genre Prediction page

Deployment with Minikube

The classification system is deployed using Minikube, a tool that enables the management and orchestration of Kubernetes clusters locally. Leveraging Minikube allows for the creation of a local Kubernetes cluster, providing a lightweight yet powerful environment for deploying and scaling containerized applications. By containerizing the system components and defining Kubernetes manifests, such as Deployment and Service configurations, the system achieves portability, scalability, and reliability across different environments. Minikube's ease of use and comprehensive feature set make it an ideal choice for local development, testing, and deployment of the classification system, ensuring consistency and efficiency in the deployment process.

V. Results

Accuracy in both training and validation is computed during model training. In addition, calculations for training loss and validation loss are made in relation to epochs. The accuracy is calculated with the formula that is mentioned below—

$$\text{Accuracy} = \frac{\textit{Correctly Classified}}{\textit{Total no of Musics}} \text{ x } 100$$

Check the Test Accuracy

The model will be trained and evaluated on the validation set before its accuracy is evaluated on the test set. The model's performance will be assessed in part by measuring the accuracy on the test set. We will now assess the model using test data. On the training dataset, we achieved an accuracy of about 97%, and on the test data, 89%.

The accuracy and loss graphs are shown below

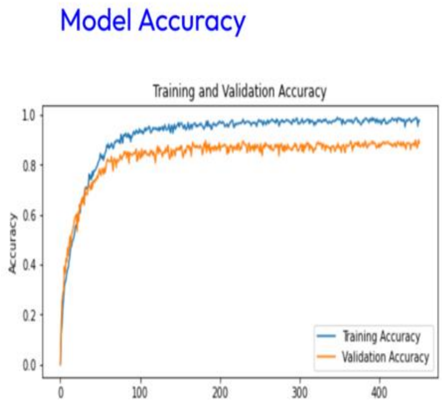

Fig.6 Model Accuracy

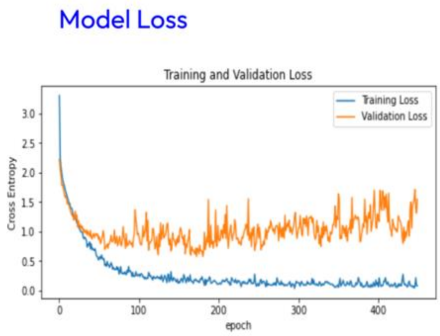

Fig.7 Model Loss

VI. Conclusion and Future Work

In this study, we have used Artificial Neural Networks (ANN) to construct a sophisticated system for classifying music genres. By improving genre classification's precision, effectiveness, and adaptability, the suggested approach seeks to give users a solid resource for categorizing and comprehending music according to its genre attributes. In comparison to the current system, the suggested method delivers better accuracy in music genre classification thanks to the use of ANN models. The system exhibits a 97% training accuracy and an 89% validation accuracy, indicating its capacity to effectively comprehend and classify a wide range of musical genres. The versatility and adaptability of the suggested approach allow it to handle a wide range of music genres and datasets. It's simple to expand and change to fit growing music libraries or new datasets, making it versatile and suitable for various music classification scenarios. Furthermore, users will have easy access to and utilization of the system thanks to its implementation on the Flask web framework. Researchers and music lovers can utilize the system because of its user-friendly interface, which enables users to upload music tracks and receive precise genre classifications. In summary, the suggested artificial neural network-based music genre classification system is a major advancement over the current one. Through the utilization of ANN models, thorough feature extraction, and effective model training, the system provides a precise, effective, and intuitive approach to categorizing and comprehending music according to its genre attributes. It creates opportunities for more research and use in the fields of recommendation systems and music analysis.

FUTURE WORK:

While there have been substantial gains with the proposed Artificial Neural Network (ANN) based music genre classification system, there are still various areas that could be improved and worked on in the future. Among them are:

- Handling unbalanced datasets: In real-world contexts, datasets pertaining to musical genres frequently experience class imbalance, with some genres being represented by noticeably more samples than others. Subsequent research endeavours may concentrate on formulating strategies to tackle the problem of class inequality, guaranteeing equitable portrayal and precise categorization of marginalized categories.
- Transfer learning and pretraining: To increase the efficacy and efficiency of the genre categorization system, pretraining methods and transfer learning may be used. When training on particular genre classification tasks, pretraining the ANN model on a large-scale music dataset or related tasks might give it a head start, resulting in faster convergence and better performance.
- Multi-label genre classification: At the moment, the suggested method concentrates on single-label genre classification, giving each musical track a unique genre label. A music track may be connected to more than one genre through multi-label genre classification, a feature that will be added to the system in future work. In order to accommodate the multiplicity and overlapping nature of musical styles, this would offer a more nuanced and thorough portrayal of music genres.
- User choices and feedback: Including user preferences can improve the system's capacity to offer tailored genre recommendations. Subsequent research endeavours may investigate methodologies such as user profiling, collaborative filtering, or active learning to modify the system's suggestions in response to specific user comments and preferences.
- Real-time classification and scalability: Enhancing the system's performance to allow for the classification of genres in real-time may prove to be a worthwhile avenue for future research. The system can be made more suited for real-time applications and large-scale music collections by lowering the computational needs and latency through the optimization of the model architecture, the use of parallel processing techniques, or the investigation of lightweight neural network models.
- Cross-cultural genre classification: By adding this feature, the system's applicability and cultural inclusivity can be increased. To completely capture the subtleties and traits unique to each cultural context, carefully selected datasets and customized models would be needed to incorporate a variety of musical traditions and genres from other cultures.

In summary, future work could concentrate on adding more features, correcting class imbalance, investigating transfer learning approaches, extending to multi-label classification, including user feedback, enhancing real-time classification, and expanding to cross-cultural genre classification. These approaches would improve the accuracy, adaptability, and utility of the music genre classification system in a variety of musical situations while also fostering new developments and applications.

REFERENCES

[1] Tzanetakis, George & Cook, Perry. (2002). Musical Genre Classification of Audio Signals. IEEE Transactions on Speech and Audio Processing. 10. 293-302.

[2] S. Vishnupriya and K. Meenakshi, "Automatic Music Genre Classification using Convolution Neural Network," 2018 International Conference on Computer Communication and Informatics (ICCCI), 2018, pp.

[3] Matthew Creme, Charles Burlin, Raphael Lenain: Music Genre Classification

[4] Zhengxin Qi, Mohamed Rahouti, Mohammed A. Jasim, and Nazli Siasi. 2022. Music Genre Classification and Feature Comparison using ML. In 2022 7th International Conference on Machine Learning Technologies (ICMLT) (ICMLT 2022). Association for Computing Machinery, New York, NY, USA,

[5] Kumaraswamy, Balachandra & Shukla, Tushar & Swati, & Satyam, Kumar. (2021). Music Genre Classification for Indian Music Genres. International Journal for Research in Applied Science and Engineering Technology.

[6] T. Feng. Deep learning for music genre classification. 2014.

[7] Archit Rathore, Margaux Dorido: Music Genre Classification.

[8] Kostrzewa, D., Kaminski, P., Brzeski, R. (2021). Music Genre Classification: Looking for the Perfect Network. In: Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds) Computational Science – ICCS 2021. ICCS 2021. Lecture Notes in Computer Science (), vol 12742. Springer, Cham.

[9] Brian McFee, Alexandros Metsai, Matt McVicar, Stefan Balke, Carl Thomé, Colin Raffel, Frank Zalkow, Ayoub Malek, Dana, Kyungyun Lee, Oriol Nieto, Dan Ellis, Jack Mason, Eric Battenberg, Scot.