WEB-BASED MUSIC GENRE CLASSIFICATION FOR TIMELINE SONG VISUALIZATION AND ANALYSIS

Dr.Maria manuel vinnay[1], J.Pujitha[2] , Abdur Faisal[3], Devarashetti Ganesh[4], Gaddameedhi Manjunath[5], Haravath Babu[6], Idupulapati Manoj[7]

[1]Professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

[2] Assistant Professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

[3,4,5,6,7] IVth Btech Student, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

ABSTRACT

This paper presents a web application that retrieves songs from YouTube and classifies them into music genres. The tool explained in this study is based on models trained using the musical collection data from Audioset. For this purpose, we have used classifier from distinct Machine Learning paradigms: Probabilistic Graphical Models (Naive Bayes), Feed-forward and Recurrent Neural Networks and Support Vector Machines (SVMs). All these models were trained in a multi-label classification scenario. Because genres may vary along a song's timeline, we perform classification in chunks of ten seconds. This capability is enabled by Audioset, which offers 10-second samples. The visualization output presents this temporal information in real time, synced with the music video being played, presenting classification results in stacked area charts, where scores for the top-10 labels obtained per chunk are shown. We briefly explain the theoretical and scientific basis of the problem and the proposed classifier. Subsequently, we show how the application works in practice, using three distinct songs as cases of study, which are then analyzed and compared with online categorizations to discuss models performance and music genre classification challenges.

1. INTRODUCTION

Research in Music Information Retrieval (MIR) comprises a broad range of topics including genre classification, recommendation, discovery and visualization. In short, this research line refers to knowledge discovery from music and involves its processing, study and analysis. When combined with Machine Learning techniques, we typically try to learn models able to emulate human abilities or tasks, which, if automated, can be helpful for the final user. Computational algorithms and models have even been applied for music generation and composition. Music genre classification (MGC) is a discipline of the music annotation domain that has recently received attention to from the MIR research community, especially since the seminal study of Tzanetakis and Cook. The main objective in MGC is to classify a musical piece into one or moremusical genres.

Music genre classification is a web application that retrieves songs from online platforms like YouTube and classier them into music genres. For this purpose, we have used classifier from distinct Machine Learning paradigms: Probabilistic Graphical Models (Naive Bayes), Feed-forward and Recurrent Neural Networks and Support Vector Machines (SVMs). All these models were trained in a multi-label classification scenario. The music classification classifies the music and group them to various genres and recommend them to the users based on the user tastes of music. This classifiers can be used on various online platforms such as spotify, YouTube etc, where we can get the favorite music based on users taste. These classifiers also help the online platform to catch the user interest and help to retain the user.

In an effort to provide a tool that gives more insights about how each genre is perceived, we have trained several classification models and embedded them in a web application that allows the user to visualize how each model ``senses" music in terms of music genre, at particular moments of a song. These models have been built using common machine learning techniques, namely, Support Vector Machines (SVM), Naive Bayes classifier, Feed forward deep neural networks and Recurrent neural networks. Whereas Bayesian and SVM methods have historically delivered good results as general purpose machine learning models, the results achieved with deep learning techniques in artificial perception (artificial vision, speech recognition, naturallanguage processing, among others) have delivered remarkable results, approaching human-like accuracy. By comparing deep learning with more traditional machine learning techniques, we also aim to compare its performance for music genre classification.

By using various classification models based on the machine learning technique like a Support Vector Machines (SVM), Naive Bayes classifier, Feed forward deep neural networks and Recurrent neural networks, it is also very difficult to classify the music based on the genres very precisely. So in order to calculate the genres more precisely, we need more advanced classification models.

The main objective in Music genre classification is to classify a musical piece into one or more musical genres. As simple as it sounds, the field still presents challenges related to the lack of standardization and vague genre definitions. The Public databases and ontologies do not usually agree on how each genre is defined. Moreover, human music perception, subject to opinions and personal experiences, makes this agreement even more difficult. For example, when a song includes swing rhythms, piano, trumpets and improvisation, we would probably define it as jazz music.

However, if we introduce synthesizers in the same song, should the song be classified as electronic music as well? If we only consider acoustic characteristics, the answer is probably yes. But different listeners can perceive the piece from their own perspective. Whereas some might categorize the song as jazz, others might consider it electronic music or even a combination of both.

1.1 Existing System

We have used classifiers from distinct Machine Learning paradigms: Feed forward and Recurrent Neural Networks, Probabilistic Graphical Models (Naive Bayes), and Support Vector Machines (SVMs). All these modelswere trained in a multi-label classification scenario. Because genres may vary along a song's timeline, we perform classification in chunks of ten seconds. This capability is enabled by Audioset, which offers 10-second samples. The visualization output presents this temporal information in real time, synced with the music video being played, presenting classification results in stacked area charts, where scores for the top-10 labels obtained perchunk are shown. We briefly explain the theoretical and scientific basis of the problem and the proposed classifier. Subsequently, we show how the application works in practice, using three distinct songs as cases of study, which are then analyzed and compared with online categorizations to discuss models performance and music genre classification challenges.

1.2 Drawbacks in Existing System

The nebulous definitions and overlapping boundaries of genres makes reliable and consistent genre classification a difficult task for humans and computers alike. Traditional rules-based classification systems are severely limited by these factors as well as by the dynamic nature of genres. The techniques used in this proposed music genre classification system are presented as alternative methodsthat can help to overcome these limitations. Arriving at a realistic and useful musical taxonomy can also be a difficult task. The problems associated with this task are briefly reviewed and some possible ways in which technology can be applied to improve the process of taxonomy construction arepresented.

1.3 Proposed System

In this paper author is using various machine learning algorithms such as Linear SVM and Ensemble Decision Tree and have also experiment with deep learning algorithms such as Feed Forward Neural Networks and LSTM (long short term memory) to classify music genre (type of music like HIP HOP, JAZZ, Disco or etc. In all algorithms LSTM is giving better accuracy. To implement this project author has used YouTube data-set called Audioset and we are also using same data-set to implement this project

1.4 Features of the Project

Here several trained classification models and embedded them in a web application that allows the user to visualize how music is classified into genre. These models have been built using common machine learning techniques namely, Support Vector Machines, Naive Bayes classifier, Feed forward deep neural networks and Recurrent neural networks, the results achieved with deep learning techniques in artificial perception have delivered remarkable results, approaching human-like accuracy. By comparing deep learning with more traditional machine learning techniques, we also compare its performance for music genre classification.By comparing deep learning with more traditional machine learning techniques, we also compare its performance for music genre classification.

2. LITERATURE SURVEY

Many researchers have worked on researching musical parameters and methods for classifying them into different genres. They have used a variety of approaches to try to replicate these skills, with varying degrees of success. The use of deep learning, particularly convolutional networks (CNNs),has lately been used successfully in computer vision and speech recognition. In Music Information Retrieval (MIR), Using audio spectrogram and MFCC, a Deep Neural Network (DNN) was developed to improve music genre classification performance.

In 2002, Musical genre classification of audio signals showed categorical labels created by humans to characterize pieces of music. A musical genre is characterized by the common characteristics shared by its members. These characteristics typically are related to the instrumentation, rhythmic structure, and harmonic content of the music. Genre hierarchies are commonly used to structure the large collections of music available on the Web. Currently musical genre annotation isperformed manually. Automatic musical genre classification can assist or replace the human user in this process and would be a valuable addition to music information retrieval systems.

In 2013, Roberto Raieli, in the Survey of Music Information Retrieval systems, presented at the Sixth International Conference on Music Information Retrieval illustrate a summary of ‘Music Information Retrieval (MIR)’, a distinction is made between the content-based search systems of general ‘audio data’ and search systems for ‘music based on the notes’.

In 2017, The bach doodle created by Cheng-Zhi Anna Huang, Curtis Hawthorne, Adam Roberts, Monica Dinculescu, James Wexler, Leon Hong, Jacob Howcroft. To make music composition more approachable, they designed the first AI-powered Google Doodle, the Bach Doodle, where users can create their own melody and have it harmonized by a machine learning model Coconet (Huang et al., 2017) in the style of Bach.

In 2020, Jean-Pierre Briot, Gaëtan Hadjeres, François-David Pachet in a paper named Deep learning techniques for music generation A survey is a survey and an analysis of different ways of using deep learning (deep artificial neural networks) to generate musical content. We propose a methodology based on five dimensions for our analysis:

In 2020, Piano automatic computer composition by deep learning and block- chain technology stated a survey to explore the automatic computer composition, investigate the copyright protection and management of digital music, and expand the application of deep learning and block-chain technologies in the generation of digital music works, piano composition was taken as a sample. Genre hierarchies are commonly used to structure the large collections of music available on the Web.

3. IMPLEMENTATION

Firstly, we opted to include YouTube as the audio source, given its undoubted popularity and the immense amount of music it offers. Additionally, it was considered appropriate because the dataset used for training our models, Audioset, included specifically sound files extracted from YouTube. However, this strong and innovative point in our application also involves certain restrictions, as the number of requests we can send to YouTube is limited, due to restrictions in the server. So, two main decisions were made: the length of the excerpts for genre classification is 10 seconds.

Input Information: The only input information required from the user in the front-end side is the unique identifier of a YouTube video, as a URL or a video ID. In theback-end, the application exposes an endpoint to classify 10-second segments from YouTube videos. The parameters required by this endpoint are the YouTube video ID and the segment start time. To classify the following segments, a front-end routine is scheduled to call the firstly classification back-end every 10 seconds in coordination with the video playback, specifying the video ID and the segment playing.

Audio Processing and Classification: When the back-end classification endpoint receives a request, the 10-second audio segment from the specified YouTube video is downloaded in WAV format. Next, the raw WAV audio isfirst converted to Mel Frequency Cepstral Coefficients (MFCCs) and then to 128- dimensional embeddings with the VGGish model. Note this is a necessary step as the models in this research have been trained using these features. The extracted features are then fed into the models. The data flow of the classification process is as shown in Figure.

3.1 Models used for implementation:

- Decision Trees
- Long Short-Term Memory
- Support Vector Machines (SVMs)
- Feed forward Neural Networks (FFNNs)

Decision Trees: Decision trees are an appealing option for classification. A decision tree is a collection of nodes, connected by branches, extending downwards from the root node to leaf nodes. Beginning at the root, attributes are tested at the decision nodes, with each possible outcome resulting in a branch. Decision trees can be trained with the classic ID3 algorithm and also with more complex solutions, such as C4.5 and C5.0. These algorithms use information gain measures to establish which variable is more informative in with respect to the target or class, in an incremental and recursive process. ID3 firstly generates too deep and complex trees.

Long Short-Term Memory: LSTM stands for long short-term memory networks, used in the field of Deep Learning. It is a variety of recurrent neural networks (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems. LSTM has feedback connections, i.e., it is capable of processing the entire sequence of data, apart from single data points such as images.

This finds application in speech recognition, machine translation, etc. LSTM is a special kind of RNN, which shows outstanding performance on a large variety of problems. LSTMs address this problem by introducing a memory cell, which is a container that can hold information for an extended period of time. The memory cell is controlled by three gates: the input gate, the forget gate, and the output gate. These gates decide what information to add to, remove from, and output from the memory cell.

Support Vector Machines (SVMs): Support Vector Machines generate hyperplanes that work as decision boundaries in high dimensional spaces to find optimal divisions between points of different classes. In short, they try to determine the best and broadest decision boundary between different classes. Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The objective of the SVM model algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

Feed Forward Neural Network(FFNN): A Feed Forward Neural Network is an artificial Neural Network in which the nodes are connected circularly. A feed-forward neural network, in which some routes are cycled, is the polar opposite of a Recurrent Neural Network. The feed-forward model is the basic type of neural network because the input is only processed in one direction. The data always flows in one direction. Feed forward neural networks are artificial neural networks in which nodes do not form loops. This type of neural network is also known as a multi-layer neural network as all information is only passed forward. During data flow, input nodes receive data, which travel through hidden layers, and exit output nodes. No links exist in the network that could get used to by sending information back from the output node. The architecture of a system reflects how the system is used and how it interacts with other systems and the outside world. It describes the interconnection of all the system's components and the data link between them.

3.2 Score Metrics for Implementations:

- **AUC(Area Under Curve)** - AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1). An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:1) True Positive Rate 2) False Positive Rate True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

$$TPR = TP/(TP+FN)$$

False Positive Rate (FPR) is defined as follows:

$$FPR = FP/(TP+FN)$$

- Accuracy – To calculate the accuracy of the module

$$Accuracy = \frac{True}{True+False} = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision and recall are often in tension. That is, improving precision typically reduces recall and vice versa.

AUC-ROC curve is one of the most commonly used metrics to evaluate the performance of machine learning algorithms.

3.3 Technologies required for implementation

Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

The Python interpreter and the extensive standard library are available in source(open-source) or binary form without charge for all major platforms, and can be freely distributed. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms object-oriented, imperative, functional and procedural, and has a large number of comprehensive standard library.

Advantages of Python :-

Let’s see how Python dominates over other languages.

- Extensive Libraries
- Extensible
- Embedded
- Improved Productivity
- IOT Opportunities
- Simple and Easy

Keras:

Keras is an open-source high-level Neural Network Library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It cannot handle low-level computations, so it makes use of the Back-end library to resolve it. The Backend library act as a high-level API wrapper for the low-level API, which lets it run on the TensorFLow, CNTK, or Theano. Keras can be developed in R as well as Python, suct that the code can be run with TensorFlow, Theano, CNTK, or MXNet as per the requirements.

Keras can run on CPU, NVIDEA, AMD GPU, TPU, etc. It ensures that producing models with Keras is really simple as it totally supports to run with TensorFlow serving, GPU acceleration(WebKeras, Keras.js), Android(TF,TF Lite), iOS(Native CodeML) and Raspberry Pi.

Tensor Flow:

Tensorflow is a library that is used in machine learning and it is an open- source library for numerical computations. It is used for developing machine learning applications and this library was first created by the Google brain team and it is the most common and successfully used library that provides various tools for machine learning applications. TensorFlow library is used in many companies in the industries like Airbnb. This company applies machine learning using TensorFlow to detect objects and classify the

NumPy:

NumPy is used for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows

Pandas:

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named NumPy. Pandas makes it simple to do many of the time consuming, repetitive tasks associated with working with data, including: Data cleansing, Data fill, Data normalization, Merges and joins, Data visualization, Statistical analysis, Data inspection, Loading and saving data and much more.

Matplotlib:

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

4. ARCHITECTURE

The architecture of a system reflects how the system is used and how it interacts with other systems and the outside world. It describes the interconnection of all the system's components and the data link between them. The architecture of a system reflects the way it is thought about in terms of its structure, functions, and relationships. In architecture, the term "system" usually refers to the architecture of the software itself, rather than the physical structure of the buildings or machinery. The architecture of a system reflects the way it is used, and therefore changes as the system is used.

Software design is a mechanism to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. It deals with representing the client's requirement, as described in SRS (Software Requirement Specification) document, into a form, i.e., easily implementable using programming language. The software design phase is the first step in SDLC (Software Design Life Cycle), which moves the concentration from the problem domain to the solution domain. In software design, we consider the system to be a set of components or modules with clearly defined behaviors & boundaries. During design, progressive refinement of data structure, program structure and procedural details are developed, reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design, procedural design.

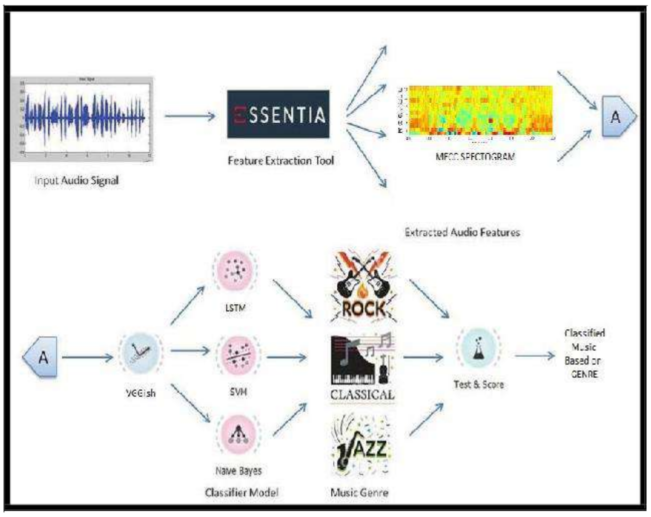

Fig 4.1: Architecture model of Web-Based Music Genre Classification

Here from the above figure, we can clearly see the architecture where when the input audio signal is provided by some website like youtube, the feature extraction tool begins and a MFCC spectrum is being made. After this, all the classifiers machine learning algorithms(Decision Trees, Naive Bayes classifier, Support Vector Machines (SVMs),Fully Connected Neural Networks (FCNNs)) are tested upon the audio signal. Various results from the machine learning models are compared with each other and the best result with utmost accuracy is being taken for the music to be classified in various genres like rock, classical, pop etc.

Here only we have to provide the input audio in the form of a link or URL and the classifier itself classifies the music into various genres. Here the extra features of this application includes the user registration, login and logout.

4.1 Modules

• **User Login:** Using this module user can login to application and after login can train with SVM, LSTM and then classify music genre

• **New User Signup Here:** Using this module user can signup with the application and then can login.

• **Train SVM:** Using this module we extract features from dataset using MFCC algorithm and this extracted features will get train with SVM and then will calculate accuracy, average precision, AUC and recall with confusion matrix graph. Here extracted features dataset will be split into train and test where 80% data used for training and 20% for testing

• **Train Decision Tree:** Using this module we extract features from dataset using MFCC algorithm and this extracted features will get train with Decision Tree and then will calculate accuracy, average precision, AUC and recall with confusion matrix graph.

• **Train LSTM:** Using this module we extract features from dataset using MFCC algorithm and this extracted features will get train with LSTM and then will calculate accuracy, average precision, AUC and recall with confusion matrix graph.

• **Train Feed Forward Network:** Using this module we extract features from dataset using MFCC algorithm and this extracted features will get train with Feed Forward Neural Network and then will calculate accuracy, average precision, AUC and recall with confusion matrix graph.

• **Music Genre Classification:** Using this module user can upload test audio files from 'testMusicFiles' folder and then LSTM will predict/classify type of that uploaded music Genre

5. FUTURE WORK

The project can be further collaborated in a web - based application or in any device supported with an in-built intelligence by virtue of Internet of Things (IoT), to be more feasible for use. Moreover, the flexibility of the proposed approach can be increased with variants at a very appropriate stage of using various models.

There is a further need of experiments for proper measurements of both accuracy and resource efficiency to assess and optimize correctly. It is believed that the extension and remnant of these metrics and matching algorithms is a promising future line of work and deserves attention.

We believe that this application could be a supporting tool for the traditional evaluation metrics in MGC, especially when manual introspection of questionable results is required beyond classic performance metrics, such as average precision or AUC. As mentioned throughout the paper, a consensus for a standardized taxonomy for music genre categorization is an open challenge for MGC. We plan to open a research line approaching this issue, and we feel we should incorporate semantic elements and ontology-based information to properly tackle the genre-mapping problem across different taxonomies.

6. CONCLUSION

The article presents a web application to discover music genres present in a song, along its timeline, based on a previous experimentation with different machine learning models. By identifying genres in each 10-second fragment, we can get an idea of how each model perceives each part of a song. Moreover, by presenting those data in a stacked area timeline graph, the application is also able to quickly show the behavior of the models, which at the same time, is an interesting way to detect undesired or rare predictions.

The article presents a web application to discover music genres present in a song, along its timeline, based on a previous experimentation with different machine learning models [6]. By identifying genres in each 10-second fragment, we can get an idea of how each model perceives each part of a song. Moreover, by presenting those data in a stacked area timeline graph, the application is also able to quickly show the behavior of the models, which at the same time, is an interesting way to detect undesired or rare predictions.

We believe that this application could be a supporting tool for the traditional evaluation metrics in MGC, especially when manual introspection of questionable results is required beyond classic performance metrics, such as average precision or AUC. It is, in any case, a challenge to establish a formal way to validate genre predictions, particularly when trying to compare them with categorizations from other sources, such as online music platforms, because there is no standard or formal way of defining genres. Last.fm, to name an example, has a completely different set of tags, which, in many cases, do not correspond or exist in the Audioset ontology.

The application is also a first step towards an eventual user-centered MGC tool, in which the users can submit feedback about the correctness of the predictions. To our knowledge, there is no visual tool that provides this level of verification on genre classification results for different fragments of the song. The design of the precision/sensitivity metric, and its use for comparing the models' results, is an additional contribution of this paper. The incorporation of available tags from public and online services enabled the proposed evaluation method.We believe that the extension and refinement of these metrics and matching algorithms is a promising future line of work and deserves attention. As mentioned throughout the paper, a consensus for a standardized taxonomy for music genre categorization is an open challenge for MGC. We plan to open a research line approaching this issue, and we feel we should incorporate semantic elements and ontology-based information to properly tackle the genre-mapping problem across different taxonomies.

Here, we compare different modules by extracting the features from dataset using MFCC algorithm and this extracted features will get train with the respective algorithms(LSTM, STM, Decision tree, FFNN) and then will calculate accuracy, average precision, AUC and recall with confusion matrix graph. Here extracted features dataset will be split into train and test where 80% data used for training and 20% for testing. Here as we can see, we get the highest precision by using the LSTM algorithm. The Accuracy of the LSTM algorithm to classify into genre is 98.75 which is the highest among the SVM, Decision Tree and the FFNN algorithms having accuracies 71.25, 47.5 and 62.5 respectively. So because of this we use the LSTM algorithm as it has the highest accuracy. So the LSTM algorithm can predict the genre of the music with utmost accuracy

7. REFERENCES

[1] J. S. Downie, ``Music information retrieval," Annu. Rev. Inf. Sci. Technol., vol. 37, no. 1, pp. 295340, 2003.

[2] H. Li, “Piano automatic computer composition by deep learning and blockchain technology," IEEE Access, vol. 8, pp. 188951188958, 2020.

[3] G. Tzanetakis and P. Cook, ``Musical genre classication of audio signals," IEEE Trans. Speech Audio Process., vol. 10, no. 5, pp. 293302, Jul. 2002.

[4] R. Basili, A. Serani, and A. Stellato, ``Classication of musical genre: A machine learning approach," in Proc. 5th ISMIR Conf., Barcelona, Spain, 2004.

[5] T. D. Nielsen and F. V. Jensen, Bayesian Networks and Decision Graphs. New York, NY, USA: Springer, 2009.

[6] D. Temperley, ``A unied probabilistic model for polyphonic music analysis," J. New Music Res., vol. 38, no. 1, pp. 318, Mar. 2009.

[7] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition," Data Mining Knowl. Discovery, vol. 2, no. 2, pp. 121167, 1998.

[8] G. E. Hinton, S. Osindero, and Y.-W. Teh, ``A fast learning algorithm for deep belief nets," Neural Comput., vol. 18, no. 7, pp. 15271554, Jul. 2006.

[9] S. Gururani, C. Summers, and A. Lerch, ``Instrument activity detection in polyphonic music using deep neural networks," in Proc. 19th ISMIR Conf., Paris, France, 2018, pp. 569576.

[10] L. R. Rabiner and B.-H. Juang, Fundamentals of Speech Recognition, vol. 14. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

[11] M. J. Flores, J. A. Gámez, and A. M. Martínez, ``Supervised classication with Bayesian networks: A review on models and applications," in Intelligent Data Analysis for Real-Life Applications: Theory and Practice. Hershey, PA, USA: IGI Global, 2012, pp. 72102.

[12] D. Temperley, ``A unied probabilistic model for polyphonic music analysis," J. New Music Res., vol. 38, no. 1, pp. 318, Mar. 2009.

[13] J. Pickens, ``A comparison of language modeling and probabilistic text information retrieval approaches to monophonic music retrieval," in Proc. 1st ISMIR Conf., Plymouth, MA, USA, 2000, pp. 111.

[14] H.-S. Park, J.-O. Yoo, and S.-B. Cho, ``A context-aware music recommendation system using fuzzy Bayesian networks with utility theory," in Proc. Int. Conf. Fuzzy Syst. Knowl. Discovery. Berlin, Germany: Springer, 2006, pp. 970979.

[15] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, ``An efcient hybrid music recommender system using an incrementally trainable probabilistic generative model," IEEE Trans. Audio, Speech, Language Process., vol. 16, no. 2, pp. 435447, Feb. 2008.

[16] S. A. Abdallah, ``Towards music perception by redundancy reduction and unsupervised learning in probabilistic models," Ph.D. dissertation, Dept. Electron. Eng., Queen Mary Univ. London, London, U.K., 2002.

[17] C. J. C. Burges, ``A tutorial on support vector machines for pattern recognition," Data Mining Knowl. Discovery, vol. 2, no. 2, pp. 121167, 1998.

[18] T. Li, M. Ogihara, and Q. Li, ``A comparative study on content-based music genre classication," in Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), 2003, pp. 282289.

[19] C. N. Silla, A. L. Koerich, and C. A. A. Kaestner, ``Improving automatic music genre classication with hybrid content-based feature vectors," in Proc. ACM Symp. Appl. Comput. (SAC), 2010, pp. 17021707.

[20] R. Dechter, ``Learning while searching in constraint-satisfactionproblems," in Proc. 5th Nat. Conf. Artif. Intell. Philadelphia, PA, USA: Morgan Kaufmann, 1986, pp. 178185.

[21] G. E. Hinton, S. Osindero, and Y.-W. Teh, ``A fast learning algorithm for deep belief nets," Neural Comput., vol. 18, no. 7, pp. 15271554, Jul. 2006.

[22] L. Deng and D. Yu, ``Deep learning: Methods and applications," Found. Trends Signal Process., vol. 7, nos. 34, pp. 197387, Jun. 2014.

[23] E. J. Humphrey, J. P. Bello, and Y. LeCun, ``Feature learning and deep architectures: New directions for music informatics," J. Intell. Inf. Syst., vol. 41, no. 3, pp. 461481, Dec. 2013.

[24] W. W. Y. Ng, W. Zeng, and T. Wang, ``Multi-level local feature coding fusion for music genre recognition," IEEE Access, vol. 8, pp. 152713152727, 2020.

[25] S. Böck, F. Krebs, and G.Widmer, ``Joint beat and downbeat tracking with recurrent neural networks," in Proc. 17th ISMIR Conf., New York City, NY, USA, 2016, pp. 255261.

[26] Y. M. G. Costa, L. S. Oliveira, and C. N. Silla, ``An evaluation of convolutional neural networks for music classication using spectrograms," Appl. Soft Comput., vol. 52, pp. 2838, Mar. 2017.

[27] R. Yang, L. Feng, H. Wang, J. Yao, and S. Luo, ``Parallel recurrent convolutional neural networks-based music genre classication method for mobile devices," IEEE Access, vol. 8, pp. 1962919637, 2020.

[28] D. H. Hubel and T. N.Wiesel, ``Receptive elds, binocular interaction and functional architecture in the cat's visual cortex," J. Physiol., vol. 160, no. 1, pp. 106154, Jan. 1962.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ``ImageNet classication with deep convolutional neural networks," in Proc. Adv. Neural Inf. Pro- cess. Syst., 2012, pp. 10971105.

[30] K. Simonyan and A. Zisserman, ``Very deep convolutional networks for large-scale image recognition," in Proc. 3rd Int. Conf. Learn. Represent., San Diego, CA, USA, 2015, pp. 114.

[31] K. He, X. Zhang, S. Ren, and J. Sun, ``Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770778.

[32] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, ``NnAudio: An on-the-Fly GPU audio to spectrogram conversion toolbox using 1D convolutional neural networks," IEEE Access, vol. 8, pp. 161981162003, 2020.

[33] G. Korvel, P. Treigys, G. Tamulevicus, J. Bernataviciene, and B. Kostek, ``Analysis of 2D feature spaces for deep learning-based speech recognition," J. Audio Eng. Soc., vol. 66, no. 12, pp. 10721081, Dec. 2018.

[34] S. Gururani, C. Summers, and A. Lerch, ``Instrument activity detection in polyphonic music using deep neural networks," in Proc. 19th ISMIR Conf., Paris, France, 2018, pp. 569576.

[35] J. S. Gómez, J. Abeÿer, and E. Cano, ``Jazz solo instrument classication with convolutional neural networks, source separation, and transfer learning," in Proc. 19th ISMIR Conf., Paris, France, 2018, pp. 577584.

[36] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, ``End-to-end learning for music audio tagging at scale," in Proc. 19th ISMIR Conf., Paris, France, 2018, pp. 637644.