RESEARCH ARTICLE OPEN ACCESS

Music genre classification using Machine learning

[1]Aliva Haiburu, [2]Swarup Bhol, [3]Mnas Ranjan Das
[1]Asst. prof., Department of CSE, Raajdhani Engineering College, Bhubaneswar [2]Metacube Software, [3]Newaetate Pvt. Ltd.

ABSTRACT
Music has always been one of the top online as well as offline usages. But not all people want to hear the same kind of rhythmic songs or genre in music. In the current world, there is a problem of automatically differentiating songs based on the user's interest. There are a few software's that classify the music genres from the data available in the details of the song, but it doesn't go through the internal rhythmic structure to classify it weather it is a rock, hip pop, etc. In order to overcome this complexity, we will classify the music through the help of machine learning technique and use several algorithms to classify it. In order to do that, we will classify the data into training data as well as test data. Using this we are going to classify the music genres and provide the easy way for the user to get the songs in their area of interest.
Keywords— Music genre, Machine learning, Classification algorithms, KNN, Principal component analysis, GTZAN DATASET.

Date of Submission: 20-05-2025 Date of acceptance: 30-05-2025

I. INTRODUCTION

With online music libraries that are readily available and easily accessible music material, people are finding difficult to classify the songs in their playlist based on their individual interests. One can classify and arrange the songs based on their genre. There are a few software that classify the music genres from the data available in the details of the song, but it doesn't go through the internal rhythmic structure to classify it whether it is a rock, hip pop, etc. based on the genre. In order to overcome this complexity, we will classify the music through the help of machine learning technique and use several algorithms to classify it. Using this we are going to classify the music genres and provide the easy way for the user to get the songs in their area of interest.

Let us first discover the top popular music genres prevalent in the contemporary music industry.

1. Pop
2. Rock
3. Indie Rock
4. EDM
5. Jazz
6. Country
7. Hip Hop & Rap
8. Classical Music
9. Latin Music
10. K-Pop

Nowadays, we can see this type of classification of songs based on the genre in few Audio streaming applications like Saavn, Wynk, Spotify and iTunes, Gaana etc. These types of applications are widely used because of their unique classification feature.

Model is designed based on the machine learning algorithm which uses CNN (Convolution Neural Network) which will take audio files as an input, then identifies and divide the songs based on the genre of the songs. And also uses the same training process involved in training audio signal of MEL spectrogram. This model uses time and frequency domain of audio signal feature for effective classification of songs which is taken from ML models like Logistic Regression, Random Forests and Vector Support Machines. Based on audio dataset, the models are evaluated. With the help of above proposed models we are trying to add-on some extra features into our model, so that we can build good system for songs classification. This paper work includes the followings.

- Existing system
- Problem Statement
- Proposed system
- Implementation
- Conclusion and Future work
- Reference

II. LITERATURE REVIEW

Music genre classification has been a widely studied area of research since the early days of the Internet. The research examined in this section incorporates many of the methods central to the present study. These include: extracting descriptive information from web-based music reviews in order to establish a genre, adhering to a rigid hierarchical genre structure to maintain standardized taxonomy and prevent artists from being torn between multiple meta-genres, and examining users' organizational schemes for both their physical and digital music collections.

Tzanetakis and Cook (2002)

This study suggests this problem with supervised machine learning approaches such as Gaussian Mixture model and k nearest neighbour classifiers. It introduced 3 sets of features for this task categorized as timbral structure, rhythmic content and pitch content. Hidden Markov Models (HMMs), which have been extensively used for speech recognition tasks, have also been explored for music genre classification (Scaringella and Zoia, 2005; Soltau et al., 1998). Support vector machines (SVMs) with different distance metrics are studied and compared in Mandel and Ellis (2005) for classifying genre.

Automatic music classification and summarization.

Automated classification and description of music is very helpful for indexing the music for faster retrieval and distribution of music over online. Extracting the popular themes from unstructured music data is very difficult. This work presents the usage of specific algorithms like SVM which split the music into pure and vocal based on the resulted attributes and also another algorithm is used for structuring, identifying and generating the music content automatically based on the knowledge of clustering concepts. The SVM method Automated classification and description of music is very helpful for indexing the music for faster retrieval and distribution of music over online. Extracting the popular themes from unstructured music data is very difficult. This work presents the usage of specific algorithms like SVM which split the music into pure and vocal based on the resulted attributes and also another algorithm is used for
structuring, identifying and generating the music content automatically based on the knowledge of clustering concepts. The SVM method is very efficient in music classification compared to Euclidean method. Accuracy of the music is checked by using listening methods on both the pure and vocal music list.

Music genre classification with machine learning techniques

The goal of this study is to use machine-learning techniques to predict the genres of the songs. For this function, the extraction of features is done using signal processing techniques, and then machine learning algorithms are applied with these features to make a multi-class music classification.

Genre classification of audio content using various classifiers and set of features

For classifying the genre of musical piece, it evaluates the output of various classifiers on multiple audio feature sets. We also test performance of sets of features collected through tools to minimize dimensionality for each classifier. Ultimately, we are working on moving up the accuracy of classification by combining various classifiers. The accuracy of test genre classification is approximately 80% plusmn 4.2 percent on 10 genre sets of 1000 pieces of music using a collection of different classifiers. This result is higher than 71.1 plusmn 7.3 percent which is the highest on this data collection. We often gain classification accuracy of 80 percent by using reduction in dimensionality or by putting together various classifiers. It is seen that best set of features is based on classification used.

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. Within seconds of hearing a new song one can easily recognize the timbre, distinct instruments, beat, chord progression, lyrics, and genre of the song. For machines on the other hand this is quite a complex and daunting task as the whole "human" experience of listening to a song is transformed into a vector of features about the song. Currently, machine learning algorithms haven't been able to surpass the 70% testing accuracy.

The aim of this project is to improve upon the accuracy of genre classification. We are considering a 10-genre classification problem with the following categories: classic pop and rock; classical; dance and electronics; folk; hip-hop; jazz and blues; metal, pop; punk; soul and reggae. The features we will use for classification are timbre, tempo and loudness information.

YouTube, Spotify and similar websites lie behind the motivation for this project. Streaming or broadcasting websites rely on metadata to organize their musical content for easier search and access by the users. A metadata is simply information about the song – album name, artist

name, song name, year of publication, genre, etc. While most of the information can easily be extracted from the title of the song, the genre is one of the important features that cannot be easily determined. With the explosion of the musical content online categorizing songs manually can soon become unrealistic. Automatic genre classification would make this process much easier and faster, and it would also improve the quality of the music recommendations. Finally, it will allow for local artists to reach to a greater audience on the web.

1. Dataset

Most of the time, first people split their collected data into to sample and validate datasets. This sample dataset is used for building the model effectively and this set is also used to train the model. Then the model is built and well trained. Then the model is tested with validate dataset actual values to verify the authenticity of the model. When planning the data for the training and test phases also include the same process.

In this work, we make use of Audio Set, which is a large-scale human annotated database of sounds (Gemmeke 2017). The dataset was created by extracting 10-second sound clips from a total of 2.1 million YouTube videos. The audio files have been annotated on the basis of an ontology which covers 527 classes of sounds including musical instruments, speech, vehicle sounds, and animal sounds and so on to. This study requires only the audio files that belong to the music category, specifically having one of the seven genre tags shown in Table.

	Genre	**count**
1	Pop- music	8100
2	Rock music	7990
3	Hip-op music	6958
4	Techno	6885
5	Rhythm Blues	4247
6	Vocal	3363
7	Reggae music	2997
	Total	40540

The number of audio clips in each category has also been tabulated. The raw audio clips of these sounds have not been provided in the Audio Set data release. However, the data provides the YouTubeID of the corresponding videos, along with the start and end times. Hence, the first task is to retrieve these audio files. For the purpose of audio retrieval from YouTube, the following steps were carried out:

1. A command line program called YouTube-dl (Gonzalez, 2006) was utilized to download the video in the mp4 format.
2. The mp4 files are converted into the desired wav format using an audio converter named ffmpeg (Tomar, 2006) (command line tool). Each wav file is about 880 KB in size, which means that the total data used in this study is approximately 34 GB.

2. Methodology

This section provides the details of the data preprocessing steps followed by the description of the two proposed approaches to this classification problem.

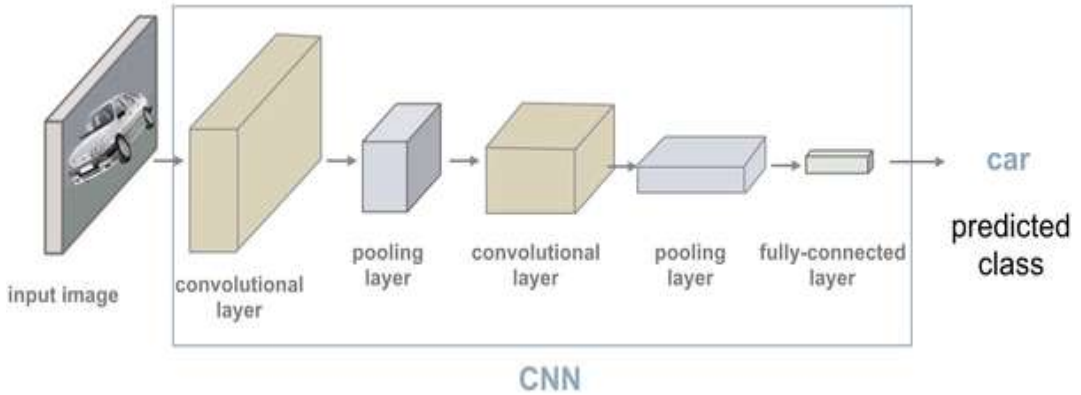

Figure 1: Convolutional neural network architecture

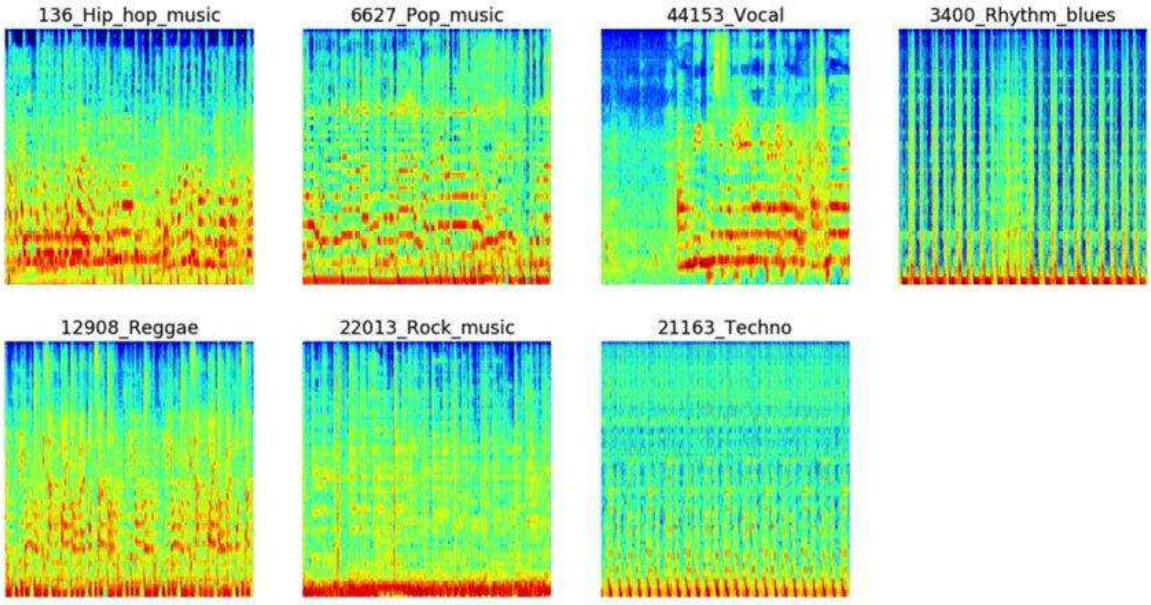

Figure 2: Sample spectrograms for 1 audio signal from each music genre

Data Pre-processing

In order to improve the Signal-to-Noise Ratio (SNR) of the signal, a pre-emphasis filter, given by Equation is applied to the original audio signal.

$$Y(t) = x(t) - \alpha * x(t-1)$$

Where, x (t) refers to the original signal, and y(t) refers to the filtered signal and α is set to 0.97. Such a pre-emphasis filter is useful to boost amplitudes at high frequencies.

Deep Neural Networks

Using deep learning, we can achieve the task of music genre classification without the need for hand-crafted features. Convolutional neural networks (CNNs) have been widely used for the task of image classification. The 3-channel (RGB) matrix representation of an image is fed into a CNN which is trained to predict the image class. In this study, the sound wave can be represented as a spectrogram, which in turn can be treated as an image. The task of the CNN is to use the spectrogram to predict the genre label (one of seven classes)

Spectrogram Generation

A spectrogram is a 2D representation of a signal, having time on the x-axis and frequency on the y-axis. A color map is used to quantify the magnitude of a given frequency within a given time window. In this study, each audio signal was converted into a MEL spectrogram (having MEL frequency bins on the y axis). The parameters used to generate the power spectrogram using STFT are listed below:

- Sampling rate (sr) = 22050
- Frame/Window size (n fft) = 2048
- Time advance between frames (hop size) = 512 (resulting in 75% overlap)
- Window Function: Hann Window
- Frequency Scale: MEL
- Number of MEL bins: 96
- Highest Frequency (f max) = sr/2

Convolutional Neural Network

There are two main parts to a CNN architecture

- A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction
- A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages as shown in Fig 1.

Convolution Layers

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers, and fully-connected (FC) layers. When these layers are stacked, a CNN architecture will be formed. In addition to these three layers, there are two more important parameters which are the dropout layer and the activation function which are defined below.

• Convolutional Layer

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken

between the filter and the parts of the input image with respect to the size of the filter (MxM).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

• Pooling Layer

In most cases, a Convolutional Layer is followed by a Pooling Layer.

The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of pooling operations.

In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer

• Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.

In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place.

• Dropout

Usually, when all the features are connected to the FC layer, it can Cause over fitting in the training dataset. Over fitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data.

To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

• Activation Functions

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network. It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax,

tanH and the Sigmoid functions. Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred for a multi-class classification, generally softmax is used.

The model consists of 5 convolutional blocks (conv. base), followed by a set of densely connected layers, which outputs the probability that a given image belongs to each of the possible classes. For the task of music genre classification using spectrograms, we download the model architecture with pre trained weights, and extract the conv base. The output of the conv base is then send to a new feed-forward neural network which in turn predicts the genre of the music, as depicted in Figure 2. There are two possible settings while implementing the pre-trained model:

1. **Transfer learning**: The weights in the convolutional base are kept fixed but the weights in the feed
forward network (represented by the yellow box in Figure 1) are allowed to be tuned to predict the correct genre label.
2. **Fine tuning**: In this setting, we start with the pre trained weights of VGG- 16, but allow all the model weights to be tuned during training process.

The final layer of the neural network outputs the class probabilities for each of the seven possible class labels. Next, the cross-entropy loss is computed as follows:

$$Loss = -\sum_{c=1}^{M} yo,c * log\, po,c \qquad (2)$$

where, M is the number of classes; yo,c is a binary indicator whose value is 1 if observation o belongs to class c and 0 otherwise; po,c is the model's predicted probability that observation o belongs to class c. This loss is used to back propagate the error, compute the gradients and thereby update the weights of the network. This iterative process continues until the loss converges to a minimum value.

III. IMPLEMENTATION

The spectrogram images have a dimension of 216 x 216. For the feed-forward network

connected to the conv. base, a 512-unit hidden layer is implemented. Over-fitting is a common issue in neural networks. In order to prevent this, two strategies are adopted:

1. L2-Regularization: The loss function of the neural network is added with the term $\frac{1}{2}\lambda \sum_i \omega_i^2$ where w refers to the weights in the neural networks. This method is used to penalize excessively high weights. We would like the weights to be diffused across all model parameters, and not just among a few parameters. Also, intuitively, smaller weights would correspond to a less complex model, thereby avoiding over fitting. λ is set to a value of 0.001 in this study.
2. Dropout: This is a regularization mechanism in which we shut off some of the neurons randomly during training. In each iteration, we thereby use a different combination of neurons to predict the final output. This makes the model generalize without any heavy dependence on a subset of the neurons. A dropout rate of 0.3 is used, which means that a given weight is set to zero during an iteration, with a probability of 0.3 is used. The dataset is randomly split into train (90%), validation (5%) and test (5%) sets. The same split is used for all experiments to ensure a fair comparison of the proposed models. The neural networks are implemented in Python using Tensor flow an NVIDIA Titan X GPU was utilized for faster processing. All models were trained for 10 epochs with a batch size of 32 with the ADAM optimizer. One epoch refers to one iteration over the entire training dataset.

Time Domain Features:

These are the features which were extracted from the row audio signal.

1. Central moments: This consists of the mean, standard deviation, skewness and kurtosis of the amplitude of the signal.
2. Zero Crossing Rate (ZCR): A zero crossing point refers to one where the signal changes sign from positive to negative. The entire 10 second signal is divided into smaller frames, and the number of zero-crossings present in each frame are determined.
3. Root Mean Square Energy (RMSE): The energy in a signal is calculated as:

$$\sum_{n=1}^{N} |x(n)|^2 \qquad (3)$$

Further, the root mean square value can be computed as:

$$\sqrt{\frac{1}{N}\sum_{n=1}^{N} |x(n)|^2} \qquad (4)$$

RMSE is calculated frame by frame and then we take the average and standard deviation across all frames.

4. Tempo: In general terms, tempo refers to the how fast or slow a piece of music is; it is expressed in terms of Beats per Minute (BPM). By intuition, different kinds of music would have different tempos. Since the tempo of the audio piece can vary with time, we aggregate it by computing the mean across several frames.

Frequency Domain Features

The audio signal can be transformed into the frequency domain by using the Fourier Transform. We then extract the following features.

1. Mel-Frequency Cepstral Coefficients (MFCC): Introduced in the early 1990s by Davis and Mermelstein, MFCCs have been very useful features for tasks such as speech recognition.
2. Chroma Features: This is a vector which corresponds to the total energy of the signal in each of the 12 pitch classes. (C, C#, D, D#, E, F, F#, G, G#, A, A#, B). The Chroma vectors are then aggregated across the frames to obtain a representative mean and standard deviation.
3. Spectral Centroid: For each frame, this corresponds to the frequency around which most of the energy is centered. It is a magnitude weighted frequency calculated as:

$$f_c = \frac{\Sigma_k S(k) f(k)}{\Sigma_k fk} \qquad (5)$$

Where S (k) is the spectral magnitude of frequency bin k and f (k) is the Frequency corresponding to bin k.

4. Spectral Contrast: Each frame is divided into a pre specified number of frequency bands. And, within each frequency band, the spectral contrast is calculated as the difference between the maximum and minimum magnitudes.

5. Spectral Roll-off: This feature corresponds to the value of frequency below which 85% of the total energy in the spectrum lies.

For each of the spectral features described above, the mean and standard deviation of the values taken across frames is considered as the representative final feature that is fed to the model.

Classifiers

This section provides a brief overview of the four machine learning classifiers adopted in this study.

1. Logistic Regression (LR): This linear classifier is generally used for binary classification tasks. For this multi-class classification task, the LR is implemented as a one-vs -rest method. That is, 7 separate binary classifiers are trained. During test time, the class with the highest probability from among the 7 classifiers is chosen as the predicted class.

2. Random Forest (RF): Random Forest is an ensemble learner that combines the prediction from a pre specified number of decision trees. It works on the integration of two main principles:

A: each decision tree is trained with only a subset of the training samples which is known as bootstrap aggregation

B: each decision tree is required to make its prediction using only a random subset of the features. The final predicted class of the RF is determined based on the majority vote from the individual classifiers.

3. Support Vector Machines (SVM): The support-vector machine builds hyper plane or collection of hyper planes in an immense- or limitless-dimensional space that is used for classification, reconstruction or other tasks such as outlier's recognition. Instinctively, the hyper plane with the greatest distance to the nearest training data point in either class achieves a strong separation, as normally the greater the difference, the less the classifier's generalization error.

Evaluation

Metrics: In order to evaluate the performance of the models described in, the following metrics will be used.

Accuracy: Refers to the percentage of correctly classified test samples.

This metric evaluates how accurate the models prediction is compare to the data.

F-score: Based on the confusion matrix, it is possible to calculate the precision and recall. F-score is then computed as the harmonic mean between precision and recall.

AUC: This evaluation criteria known as the area under the receiver operator characteristics (ROC) curve is a common way to judge the performance of a multi-class classification system. True positive rate and the false positive rate.

IV. RESULTS AND DISCUSSION

In this section, the different modelling approaches discussed in Section 5 are their accuracies.

These accuracies are shown in Table below.

	Accuracy	**F score**	**AUC**
Spectrogram-based models			
CNN Transfer Learning	0.63	0.61	0.891
CNN Fine Tuning	0.64	0.61	0.889
Feed-Forward NN baseline	0.43	0.33	0.759
Feature based models			
Logistic Regression(LR)	0.53	0.47	0.822
Random Forest(RF)	0.54	0.48	0.840
Support Vector Machines(SVM)	0.57	0.52	0.856

Comparison of performance of the models on the test set

The best performance in terms of all metrics is observed for the convolutional neural network model based on VGG-16 that uses only the spectrogram to predict the music genre. It was expected that the fine tuning setting, which additionally allows the convolutional base to be trainable, would enhance the CNN model when compared to the transfer learning setting. The

experimental results show that there is no significant difference between transfer learning and fine-tuning. The baseline feed forward neural network that uses the unrolled pixel values from the spectrogram performs poorly on the test set. This shows that CNNs can significantly improve the scores on such an image classification task. The models that use manually crafted features, the one with the least performance is the Logistic regression model. This is expected since logistic regression is a linear classifier. SVMs outperform random forests in terms of accuracy.

V. CONCLUSION AND FUTURE WORK

This program falls under nowadays sophisticated methodology of machine learning. RF, SVM is better suited for numerical processing especially in classification. In conclusion, as described through the literature review, we conclude that only a marginal progress is achieved in developing a predictive model for classification of Music Genres, and hence the need for combinational and more complex models to improve the accuracy of classification of the music. We conclude that the experimental outcome is more reliable than what we get from the existing method.

In the future, we hope to experiment with other types of deep learning methods, given they performed the best. Given that this is time series data, some sort of RNN model may work well (GRU, LSTM, for example). We are also curious about generative aspects of this project, including some sort of genre conversion (in the same vein as generative adversarial networks which repaint photos in the style of Van Gogh, but for specifically for music). Additionally, we suspect that we may have opportunities for transfer learning, for example in classifying music by artist or by decade.

REFERENCES:

[1]. Gerald Penn, Abdel-Rahman Mohamed, Li Deng, Hue Jiang, Dong Yu and Osama Abdel- Hamid. August 2014. Convolutionary spoken word comprehension networks. IEEE / ACM Audio, voice and language management teams 22(10):1533- 1545

[2]. G. Tzanetakis, and C .Perry. "Musical genre classification of audio signals." IEEE Transactions on speech and audio processing 10, no. 5, pp. 293-302, 2002.

[3]. Hareesh Bahuleyan, Music Genre Classification using Machine Learning Techniques, University of Waterloo, 2018.

[4]. Tom LH Li, Antoni B Chan, and A Chun. Automatic musical pattern feature extraction using convolutional neural network. In Proc. Int. Conf. Data Mining and Applications, 2010.

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

[6]. Fu, Z., Lu, G., Ting, K. M., & Zhang, D. A survey of audio based music classification and annotation. IEEE Transactions on Multimedia, 13(2), 303–319, 2011.

[7]. G Tzanetakis and P Cook. Musical genre classification of audio signals. IEEE Trans. on Speech and Audio Processing, 2002.

[8]. S. Lippens, J.P Martens, T. De Mulder, G. Tzanetakis. A Comparison of Human and Automatic Musical Genre Classification. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. 1520- 6149, IEEE.

[9]. Lonce Wyse, Audio Spectrogram representations for processing with Convolutional Neural Networks, National University of Singapore, 2017.

[10]. Dan Ellis. 2007. Analysis and synthesis of the Chroma functions. Laboratory tools for Speech Recognition and Organization and Audio-LabROSA.