ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

An Identification of Musical Genre: Sarangi as a Special case

Madhuri R. Desai[1]

Assistant Professor, Department of Electronics and Tele-Communication Engineering, Sanjay Ghodawat University, Maharashtra, India[1]

ABSTRACT: Musical genres are defined as categorical labels that auditors use to characterize pieces of music. The musical genre can be characterized by a set of common parameters shared by its members. These parameters are closely related to the instrumentation, rhythmic structure and also harmonic content of the music. An automatic genre classification would actually be very helpful to replace or complete human genre annotation, which is actually used. In this paper, we explore the automatic signal identification of a musical sarangi database. More specifically, 13 coefficients of MFCC feature are used to identify the different parameters of sarangi. The automatic identification of this database is then evaluated through an artificial neural network; the parameters of this are optimized to obtain the best scores in each case. Interesting comparative results are reported and commented In addition, automatic musical genre classification provides a framework for developing and evaluating features for any type of content- based analysis of musical signals.

KEYWORDS: Audio classification, feature extraction, musical genre.

I. Introduction

Audio signal classification system analyses the input audio signal and creates a label that describes the signal at the output. These are used to characterize both music and speech signals. The categorization can be done on the basis of pitch, music content, music tempo and rhythm. Audio signal classification finds its utility in many research fields such as audio content analysis, broadcast browsing, and information retrieval. All classification systems employ the extraction of a set of features from the input signal. Each of these features represents an element of the feature vector in the feature space. The dimension of the feature space is equal to the number of extracted features. These features are given to a classifier that employs certain rules to assign a class to the incoming vector. Fig.1.1 shows the block diagram, which is self-explanatory [1].

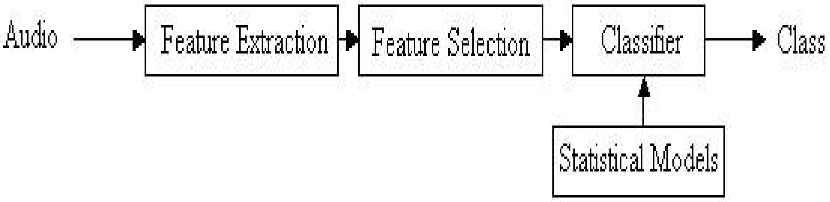

Fig 1.1 audio signal classification

A FEATURE EXTRACTION

Feature extraction is the process of computing a compact numerical representation that can be used to characterize a segment of audio. The design of descriptive features for a specific application is the main challenge in building pattern recognition systems. Once the features are extracted standard machine learning techniques which are independent of the specific application area can be used. Before any audio signal can be classified under a given class, the features in that audio signal are to be extracted. These features will decide the class of the signal. Feature extraction involves the analysis of the input of the audio signal. The feature extraction techniques can be classified as temporal analysis and spectral analysis technique. Temporal analysis uses the waveform of the audio signal itself for analysis. Spectral

DOI: 10.15680/IJIRCCE.2017. 0511070

analysis utilizes spectral representation of the audio signal for analysis. All audio features are extracted by breaking the input signal into a succession of analysis windows or frames, each of around 10-40-ms length, and computing one feature value for each of the windows. One approach is to take the values of all features for a given analysis window to form the feature vector for the classification decision, so that class assignments can be obtained almost in real time, thus realizing a real-time classifier. Another approach is to use the texture window, in which the long-term characteristics of the signal are extracted and the variation in time of each feature is measured, that often provides a better description of the signal than the feature itself [1].

A texture window is a long-term segment in the range of seconds containing a number of analysis windows. In the texture based approach only one feature vector for each texture window is generated. The features are not directly obtained in each analysis window, but statistical measures of the values are obtained for all analysis windows within the current texture window. Therefore in this case real-time classification is not possible, since at least one whole texture window has to be processed to obtain a class decision. Since the analysed audio files are supposed to contain only one type of audio, a single class decision is made for each type of audio, which can be derived following one of two possible approaches. The first approach is the single vector mode, which consists of taking the whole file length as the texture window. In this way, each file is represented by a single feature vector, which in turn is subjected only once to classification. The second approach is the texture window mode, which consists of defining shorter texture windows and making several class decisions along each file, one for each texture window. At the end of the file the decisions are averaged to obtain a final class decision. This average computation is weighted by the certainty of each class decision.

B FEATURE SELECTION

From a large set of features it is important to select particular set of features that would determine the nature and hence the class of the audio signal. These features determine the dimensionality in the feature space. It is important therefore to select an optimum number of features that not only keeps accordance with the accuracy and the level of performance but also reduces the computation costs. Thus there is no point in increasing the number of features as it would not have a drastic impact on the accuracy but would pave for more complexities in computation [2].

C CLASSIFIER

After the feature selection process it is important to classify the signal. Classification is the process by which a particular label is assigned to a particular audio format. It is this label that would define the signal and its origin. A classifier defines decision boundaries in the feature space (i.e. mean vs. maximum), which separate different sample classes from each other. Classifiers are categorized by their real time capabilities, on the basis of the approach and their character. On the basis of their real time capabilities, there are real time classifiers and non-real time classifiers. Real time classifiers can update classification results in time intervals of milliseconds. Hence their application comes of importance in the areas where the input signal consists of a sequence of different types of audio and it is absolutely necessary to keep updating, for class detection. In case of the non-real time classifiers, they analyse a longer fragment of the signal before they provide a classification result. Accuracy in this case is more than real time classifiers because they analyse a longer fragment of the incoming signal, which plays a prominent role to describe the signal [3].

II. STRING INSRTUMENTS

For identification purpose sarangi is taken as a string instrument All signals of sarangi are recorded at 44.1KHz sampling frequency and are of same category but having different properties for example frequency, tension, density etc.

III. SYSTEM SETUP

This section describes the setup of the digital audio classification system. This system is composed primarily of the blocks above and was developed in the Matlab environment. Matlab code can be provided upon request.

A. *Input Files*

Data for training and testing the system was taken from four same types of string instruments. The tracks on each of these were extracted and converted to WAV format and then divided into segments of length 86 bits, or one seconds. To avoid periods within the music not characteristic of the whole song, the segments were all taken from the middle of each track. For classification by genre, for classification by instrument four tracks were used.

B. *Filtering Process*

Method to remove silence part from an audio signal

- Take i/p signal as ‘Y’
- Apply sampling frequency 44.1kHz to signal
- Take absolute value of i/p signal
- Decide the silence threshold level
- Detection of silence part after multiplying by each bit of signal
- Represent the silence part in red while non silence part in green
- Plot these two figures

C. *Fast Fourier Transform*

A sinusoid is a mathematical function that traces out the simplest repetitive motion in nature. A ball on a rubber band will descend and slow as the band stretches, stop when the gravitational acceleration equals the restoring force of the rubber band, begin to ascend and stop again when the restoring force is zero and the gravitational acceleration equals the momentum. This system is called a simple harmonic oscillator. The repetitive up-and-down motion that it creates is called a sine wave or a sinusoid, and is found in many different forms in nature. In particular it is found in the varying air pressure of sound waves. Any sound can be created by adding together an infinite number of these sine waves. This is the essence of Fourier synthesis. In the more general sense, any function can be generated from the summation of an infinite number of sinusoids of different frequencies and amplitudes. The frequency of a sinusoid is how many times it repeats in one second. The amplitude is how high the oscillation reaches. In our ball and rubber band example, the amplitude is the farthest up or down the ball travels from the resting state. Humans and other vertebrates have an organ called the cochlea inside the ear that analyses sound by spreading it out into its component sinusoids. One end of this organ is sensitive to low frequency sinusoids, and one end is sensitive to higher frequencies. When a sound arrives, different parts of the organ react to the different frequencies that are present in the sound, generating nerve impulses which are interpreted by the brain.

Fourier analysis is a mathematical way to perform this function. The opposite of Fourier synthesis, Fourier analysis consists of decomposing a function into its component sinusoids. The Fourier transform is a mathematical way to go between the functional representation of a signal and its Fourier representation. The Fourier representation of a signal shows the spectral composition of the signal. It contains a list of sinusoid functions, identified by frequency, and each sinusoid has an associated amplitude and phase. The phase of a signal is the start location of the sinusoid relative to some specific zero. Phase is measured as an angle, in degrees or radians, indicating some part of a complete oscillation. A sinusoid with a phase of 0 radians will be identical to a sinusoid with a phase of 2 radians. These signals are said to be “in phase”. A sinusoid with a phase of radians is the numerical opposite of a sinusoid with a phase of 0 radians. These signals are said to be “out of phase” and if combined, would cancel each other out. It has been shown that the ear is “phase deaf”, which means that two sinusoids with different phases will be perceived as the same sound. In fact, two spectrally rich sounds with all frequency components having different phases, as in Figure 2, will sound the same. For this reason, the phase component of the Fourier representation is often discarded. However it has also been shown that while two steady state signals with the same amplitude spectrum sound the same regardless of their phase spectra, changes in the phase spectrum of a signal over time are perceivable. This change in phase is perceived as a shift in timbre, but not in pitch, so the phase information may be important depending on the application [4].

D. *MFCC- Mel Frequency Cepstral Coefficient*

MFCCs are primarily used as features for speech and speaker recognition. Many have used MFCCs to model music and audio signals. MFCCs are short-term spectral features that are obtained from a type of cepstral representation of the audio segment. The main difference between the normal cepstrum and the MFCC is that in MFCC, the frequency bands are positioned logarithmically or in mel-scale, which represents the human auditory system's response more closely than the linearly-spaced frequency bands obtained directly from using the ordinary Fast Fourier Transform (FFT) or Discrete Cosine Transform (DCT). Audio files usually have both speech and music components, which tend to make the feature extraction task difficult and MFCCs are primarily used to circumvent this problem. In certain genres music may be more predominant than vocals (example Classical, where vocal(s) if present are less speech like, due to high pitch) and in certain cases it might be just the reverse (example Rap), in such cases MFCCs can work as a crucial discriminatory feature. It is well known that the first 13 MFCC coefficients provide the best speaker related information [7]. The third feature vector (V) is based upon the first 13 MFCC coefficients. As these coefficients retain speaker related information, they can be effectively used to classify audio samples according to vocalist(s). For each window 13 MFCC coefficients were calculated, generating a total of 263 sets of 13 MFC-coefficients. Each MFC-coefficient has a vector of 263 values, corresponding to 263 windows; hence the overall MFC-coefficient set has a dimension of 13x263.

MFCC’s employ the mel scale which is a scale of pitches which are equal in distance from one another. The normal frequency f hertz can be converted to the mel range by the following equation

$$\mathbf{Fm=2595*(log\ (1+ (Z1/700)));}$$

A cepstrum is the result of taking the Fourier transform of the decibel spectrum (power spectrum) as if it were a signal. There is a complex cepstrum and a real cepstrum. The cepstrum can be defined mathematically as cepstrum of a signal = FT (log (FT (the signal))) where FT indicates Fourier Transform. The real cepstrum uses the logarithm function defined for real values, while the complex cepstrum uses the complex logarithm function defined for complex values. The complex cepstrum holds information about magnitude and phase of the initial spectrum, allowing the reconstruction of the signal. The real cepstrum only uses the information of the magnitude of the spectrum. The cepstrum can be seen as information about rate of change in the different spectrum bands. Usually the spectrum is first transformed using the mel frequency bands. The result is called the MFCC's, which are used for voice identification, pitch detection and much more. This is a result of the cepstrum separating the energy resulting from vocal cord vibration from the "distorted" signal formed by the rest of the vocal tract. The human ear exhibits a nonlinear characteristic when it comes to the perception of pitch. Hence the mel scale takes into the account of this property. Below 500Hz the frequency and the mel scales coincide and above that larger and larger intervals produce equal pitch increments. As a result, four octaves on the hertz scale above 500Hz are judged to comprise about two octaves on the mel scale. After the translation to the mel frequency scale the coefficients can be evaluated. Normally the computation of MFCC s involves the windowing of the incoming audio signal. The log of the spectrum is computed and another transform is applied in order to obtain the cepstrum coefficients. This can be explained from Fig.3.1 as follows

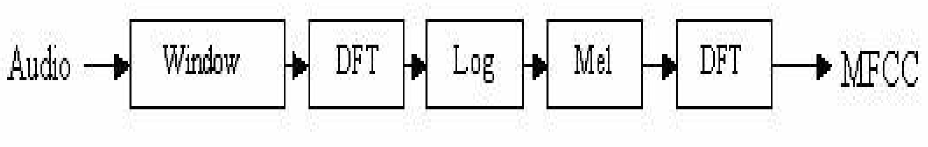

Fig.3.1 Block diagram to compute MFCC’s

First the audio is hamming windowed in overlapping steps. For each window, the log of the power spectrum is computed using a DFT. A nonlinear map of the frequency scale perceptually weights the log spectral coefficients. This operation called the mel scaling, emphasizes mid frequency bands in proportion to their perceptual importance. At the final stage the mel weighted spectrum is transformed into cepstral coefficients using another DFT. This results in features that are dimensionally uncorrelated. Thus MFCC’s provide a compact representation of the spectral envelope, such that most of the signal energy is concentrated in the first few coefficients. MFCC’s were originally invented for characterizing the seismic echoes resulting from earthquakes and bomb explosions. It is now used as an excellent feature vector for representing the human voice and musical signals. We will give a high level intro to the

implementation steps, then go in depth why we do the things we do. Towards the end we will go into a more detailed description of how to calculate MFCCs.
Apply a window function (e.g. the Hamming window) and compute the discrete Fourier transform.

- Group the frequency bins into M bins equally spaced on the mel frequency scale with 50% overlap.
- Take the logarithm of the sum of each bin.
- Compute the discrete cosine transform of the logarithms.
- Discard high-frequency coefficients from the cosine transform.

E. *DCT- Discrete Cosine Transform*

A technique for converting signal into elementary frequency components

F. *ANN-Artificial Neural Network*

One type of network had shown in fig 3.2, the nodes as ‘artificial neurons’. These are called artificial neural networks (ANNs). An artificial neuron is a computational model inspired in the natural neurons. Natural neurons receive signals through synapses located on the dendrites or membrane of the neuron. When the signals received are strong enough, the neuron is activated and emits a sign although the axon. This signal might be sent to another synapse, and might activate other neurons.

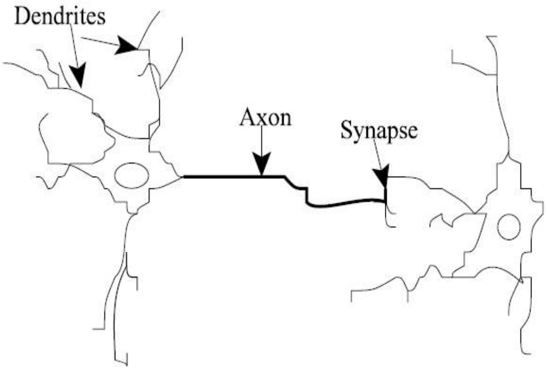

Fig 3.2 Natural neurons (artist’s conception)

The complexity of real neurons is highly abstracted when modelling artificial neurons. These basically consist of inputs (like synapses), which are multiplied by weights (strength of the respective signals), and then computed by a mathematical function which determines the activation of the neuron. Another function (which may be the identity) computes the output of the artificial neuron (sometimes in dependence of a certain threshold). ANNs combine artificial neurons in order to process basic information. Fig 3.3 shows a simplified flow for an artificial neural network.

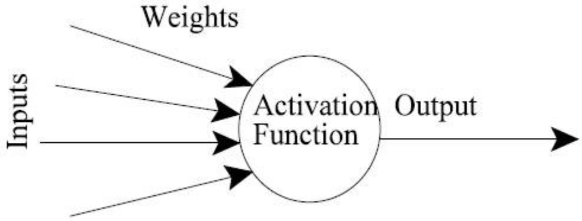

Fig 3.3 an artificial neuron

The higher a weight of an artificial neuron is, the stronger the input which is multiplied by it will be. Weights can also be negative, so we can say that the signal is inhibited by the negative weight. Depending on the weights, the computation of the neuron will be different. By adjusting the weights of an artificial neuron we can obtain the output we want for specific inputs.

But when we have an ANN of hundreds or thousands of neurons, it would be quite complicated to find by hand all the necessary weights. But we can find algorithms which can adjust the weights of the ANN in order to obtain the desired output from the network. This process of adjusting the weights is called learning or training. The number of types of ANNs and their uses is very high. The development process for an ANN application has eight steps.

Step 1: (Data collection) the data to be used for the training and testing of the network are collected. Important considerations are that the particular problem is amenable to neural network solution and that adequate data exist and can be obtained.

Step 2: (Training and testing data separation) Training data must be identified, and a plan must be made for testing the performance of the network. The available data are divided into training and testing data sets. For a moderately sized data set, 80% of the data are randomly selected for training, 10% for testing, and 10% secondary testing.

Step 3: (Network architecture) network architecture and a learning method are selected. Important considerations are the exact number of perceptron and the number of layers.

Step 4: (Parameter tuning and weight initialization) There are parameters for tuning the network to the desired learning performance level. Part of this step is initialization of the network weights and parameters, followed by modification of the parameters as training performance feedback is received. Often, the initial values are important in determining the effectiveness and length of training.

Step 5: (Data transformation) transforms the application data into the type and format required by the ANN.

Step 6: (Training) Training is conducted iteratively by presenting input and desired or known output data to the ANN. The ANN computes the outputs and adjusts the weights until the computed outputs are within an acceptable tolerance of the known outputs for the input cases.

Step 7: (Testing) Once the training has been completed, it is necessary to test the network. The testing examines the performance of the network using the derived weights by measuring the ability of the network to classify the testing data correctly. Black-box testing (comparing test results to historical results) is the primary approach for verifying that inputs produce the appropriate outputs.

Step 8: (Implementation) Now a stable set of weights are obtained. Now the network can reproduce the desired output given inputs like those in the training set. The network is ready to use as a stand-alone system or as part of another software system where new input data will be presented to it and its output will be a recommended decision [8].

IV. Results

Trained Network: After training 15 audio signals file we get the following trained file output having 1 to 15 columns

TABLE I

Trained files

Columns 1 through 8							
0.0007	0.0045	0.0036	-0.0002	0.0180	0.1039	0.1133	0.1043
Columns 9 through 15							
-0.2967	0.1039	0.3084	-0.0226	0.0340	0.2016	0.1672	

TABLE III

ANN results for the 5 different Sarangi signals

Name of audio file	Actual audio	Value of classifier	Identified audio using ANN
Sarangi1.wav	Sarangi	0.1001	Sarangi
Sarangi2.wav	Sarangi	0.1005	Sarangi
Sarangi3wav	Sarangi	0.1009	Sarangi
Sarangi4.wav	Sarangi	0.1001	Sarangi
Sarangi5.wav	Sarangi	0.0997	Sarangi

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

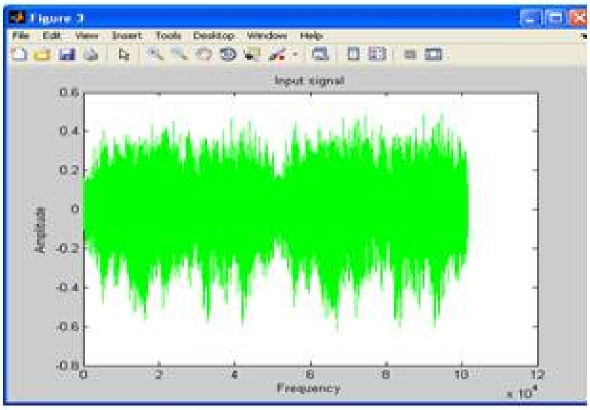

Fig 4.1 Input signal: Sarangi

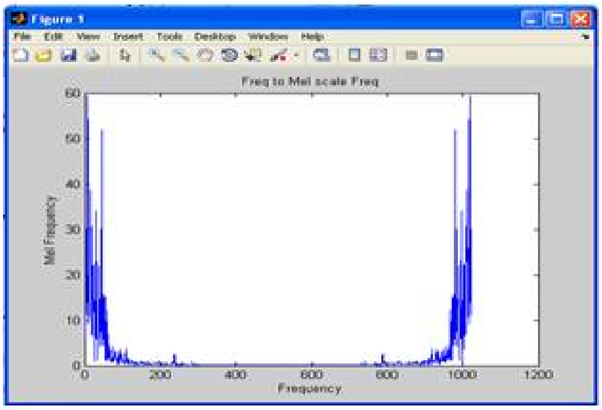

Fig 4.3 MFCC coefficients

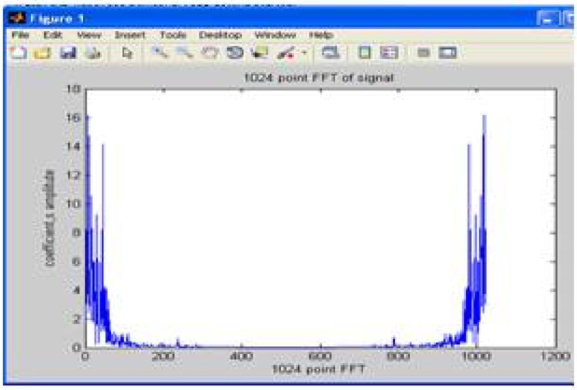

Fig 4.1 FFT of signal:

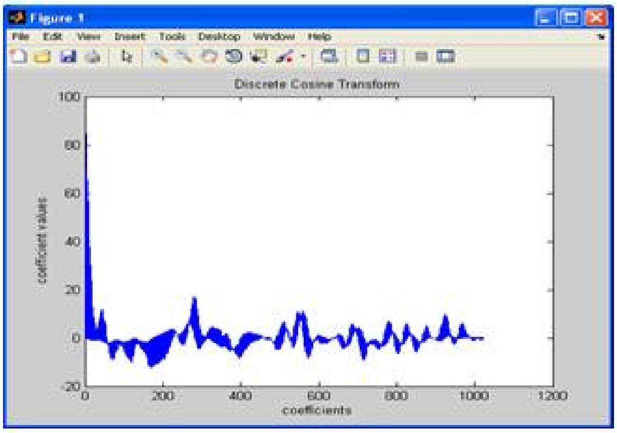

Fig 4.4 DCT of signal

V. CONCLUSION AND FUTURE WORK

Audio signals identification is very important in many audio applications so that different audio signal can be processed appropriately. We propose an audio classification scheme which will categorize string instruments based on a number of audio features. We have designed an audio classification procedure based on main characteristic of different types of sounds. Currently musical genre annotation is performed 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.

REFERENCES

1. F. Pachet and D. Cazaly, “A classification of musical genre,” in Proc.RIAO Content-Based Multimedia Information Access Conf., Paris,France, Mar. 2000.
2. A Technique towards Automatic Audio Classification and RetrievalGuojun Lu and Templar HankinsonGippsland School of Computing and Information TechnologyMonash University, Churchill, Vic 3842
3. G. Tzanetakis and P. Cook, Musical Genre Classification of Audio Signals, IEEETrans. Speech and AudioProcess, vol. 10, pp. 293 302, 2002 July.
4. J. J. Burred and A. Lerch, Hierarchical Automatic Audio Signal Classification,J. Audio Eng. Soc, Vol. 52, pp. 724-739, July/August 2004

5. J. Foote, A Similarity Measure for Automatic Audio Classification, Proc. AAAI1997 Spring Symp. on Intelligent Integration and Use of Text, Image, Video, andAudio Corpora, Stanford, CA, 1997.
6. Musical instrument recognition using ceptral coefficients and temporal features,AnttiEronen and AnssiKlapuriSignal Processing Laboratory, Tampere University of TechnologyP.O.Box 553, FIN-33101 Tampere, FINLAND
7. A. Eronen, Comparison Of Features For Musical Instrument Recognition, NewPaltz, New York, 2001 October.
8. Audio Signal Classification, Proc. AAAI1997 Spring Symp. on Intelligent Integration and Use of Text, Image, Video, andAudio Corpora, Stanford, CA, 1997.