International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163
Issue 04, Volume 5 (April 2018) www.ijirae.com

CONTENT BASED AUDIO CLASSIFIER & FEATURE EXTRACTION USING ANN TECNIQUES

K.Karthikeyan
Research Scholar, Department of Computer Science,
Marudupandiyar College, Thanjavur, Tamil Nadu, India
Dr.R.Mala
Assistant Professor, Department of Computer Science,
Marudupandiyar College, Thanjavur, Tamil Nadu, India

Manuscript History
Number: **IJIRAE/RS/Vol.05/Issue04/APAE10081**
DOI: 10.26562/IJIRAE.2018.APAE10081
Received: 02, April 2018
Final Correction: 12, April 2018
Final Accepted: 17, April 2018
Published: **April 2018**
Citation: Karthikeyan & Mala (2018). CONTENT BASED AUDIO CLASSIFIER & FEATURE EXTRACTION USING ANN TECNIQUES. IJIRAE::International Journal of Innovative Research in Advanced Engineering, Volume V, 106-116. **doi://10.26562/IJIRAE.2018.APAE10081**
Editor: Dr.A.Arul L.S, Chief Editor, IJIRAE, AM Publications, India

ABSTRACT: Audio signals which include speech, music and environmental sounds are important types of media. The problem of distinguishing audio signals into these different audio types is thus becoming increasingly significant. A human listener can easily distinguish between different audio types by just listening to a short segment of an audio signal. However, solving this problem using computers has proven to be very difficult. Nevertheless, many systems with modest accuracy could still be implemented. The experimental results demonstrate the effectiveness of our classification system. The complete system is developed in ANN Techniques with Autonomic Computing system.

Keywords: MFCC; ANN; Knowledge Base; Learning Process; Energy; Audio feature extraction;

I. INTRODUCTION

Audio segmentation and classification have applications in wide areas. For instance, content based audio classification and retrieval is broadly used in the entertainment industry, audio archive management, commercial music usage, surveillance, etc. There are many digital audio databases on the World Wide Web nowadays; here audio segmentation and classification would be needed for audio searching and indexing. Recently, there has been a great deal of interest in monitoring broadcast news programs, in this case classification of speech data in terms of speaker could help in efficient navigation through broadcast news archives.

In music psychology and music education, emotions based components of music has been recognized as the most strongly component associated with music expressivity. Music information behavior studies have also identified music mood emotion as an important criterion used by people in music seeking indexing and storage. However, evaluation of music mood is difficult to classify as it is highly subjective. Although there seems to be a very strong connectivity between the music (the audio) and the mood of a person. There are many entities which explicitly change our mood while we are listening music. Rhythm, tempo, instruments and musical scales are some such entities. There is one very important entity in the form of lyrics which directly affects our minds. Identifying audible words from lyrics and classifying the mood accordingly is a difficult problem as it includes complex issues of Digital Signal Processing.

We have explored rather a simple approach of understanding the mood on the basis of audio patterns. This problem resembles to classical problem of pattern recognition. We have made an effort to extract these patterns from the audio as audio features.

1.1 PROBLEM ANALYSIS

The process of identifying the correct genre of audio is a natural process for our mind. The Human brain processes and classifies the audios naturally, based on the long history of learning and experience. Here we emulate the same methodology (inspired by the biological nervous system) by training the system to make and use the knowledge gained to classify the Neural Networks. We have identified a set of computable audio features of audios and have developed methods to understand them. These features are generic in nature and are extracted from audio streams; therefore, they do not require any object detection, tracking and classification. These features include time domain, pitch based, frequency domain, sub band energy and MFCC as audio features. We have then developed an audio classifier that can parse a given audio clip into predefined genre categories using extracted audio features of the audio clips. The audio classifier is designed using multi layerFeedforward neural network with back propagation learning algorithm and tested the classifier for characterization of audio into sports, news and music. Music genre is further classified into three moods happy, angry and sad.

II. LITERATURE REVIEW

The auditory system of humans and animals, can efficiently extract the behaviorally relevant information embedded in natural acoustic environments. Evolutionary adaptation of the neural computations and representations has probably facilitated the detection of such signals with low SNR over natural, coherently fluctuating background noises [23, 24]. It has been argued [47] that the statistical analysis of natural sounds - vocalizations, in particular - could reveal the neural basis of acoustical perception. Insights in the auditory processing then could be exploited in engineering applications for efficient sound identification, e.g. speech discrimination from music, animal vocalizations and environmental noises. In the early 2000s, existing speech recognition research was expanded upon in hopes of finding practical music labeling applications. Commonly used metrics such as Mel-frequency cepstral coefficients (MFCCs), a succinct representation of a smoothed audio spectrum, were quickly adapted for music. These tumbrel features proved to be quite effective when applied to segmented audio samples. However, given the complexities of music, additional features were necessary for sufficient genre analysis. Rhythmic and harmonic features (e.g. tempo, dominant pitches) were developed in response. Some went a step further to utilize quantization of higher level descriptors like emotion and mood to assist in the categorization of audio. The effectiveness of these types of extracted features has led some researchers to ponder their applications for automatic playlist generation, music recommendation algorithms, or even assisting musicologists in determining how humans define genres and otherwise classify music [9]. Two further machine learning methods were compared, support vector machines (SVM) and linear discriminate analysis. Findings were positive, with an overall accuracy of 80%.

McKinney and Beebart [42] suggested the usage of psychoacoustic features, e.g. roughness, loudness, and sharpness. Additional low-level features such as RMS and pitch strength were incorporated as well. Features were computed for four different frequency bands before being applied to a Gaussian mixture model, resulting in a 74% success rate. Lidy and Rauber [67] analyzed audio from a pyscho-acoustic standpoint as well, deciding to further transform the audio in the process of feature extraction. Using SVMs with pair wise classification, they were able to reach an accuracy of 75%. Burred and Lerch [29] chose to use MPEG-7 LLDs as well, but added features like beat strength and rhythmic regularity. Using an intriguing hierarchal decision-making approach, they were able to achieve a classification accuracy of around 58%. As mentioned previously, these studies often failed to depict electronic music in a manner accurately representative of the state of the genre. If included at all, terms used ranged from “Techno” [66] and “Disco” [20] to “Techno/Dance” [29] and “Eurodance” [17]. Many of these labels are now antiquated or nonexistent, so audio features and machine learning strategies were selected from previous studies for application on samples from current electronic music subgenres that better embody the style of music. Saunders [32] published one of the first studies on speech/music discrimination in hopes of isolating music portions of FM radio broadcasts. Based on analysis of the temporal zero-crossing rate (ZCR) of audio signals, a Gaussian classifier was developed from a training set of samples. A remarkable 98.4% segmentation accuracy was achieved with real-time performance, proving not only the viability of speech/music discriminators in general, but also the effectiveness of this type of two-step feature extraction and machine learning process for signal classification purposes.

III. AUDIO FEATURE EXTRACTION

Audio features extraction is the process of converting an audio signal into a sequence of feature vectors carrying characteristic information about the signal. These vectors are used as basis for various types of audio analysis algorithms. It is typical for audio analysis algorithms to be based on features computed on a window basis. These window based features can be considered as short time description of the signal for that particular moment in time.

The performance of a set of features depends on the application. The design of descriptive features for a specific application is hence the main challenge in building audio classification systems. Audio in fact tells a lot about mood of the clip, the music component, the noise, fast or slowness of the pace and the human brain too can classify just on the base of audio.

A wide range of audio features exist for classification tasks. These fall under the following categories:

- Time Domain Features
- Pitch Based Features
- Frequency Domain Features
- Energy Features
- MFCC

3.1 TIME DOMAIN FEATURES

Volume is a reliable indicator for silence detection; therefore, it can be used to segment audio sequence and determine clip boundaries [28, 71]. Volume is commonly perceived as loudness since natural sounds have pressure waves with different amount of power to push our ear. In electronic sound, the physical quantity is amplitude, which is particularly characterized by the sample value in digital signals. Therefore volume is often calculated as the *Root-Mean-Square (RMS)* of amplitude [19, 63, 76]. Volume of the n^{th} frame is calculated, by the following formula:

$$V(n) = \sqrt{1/N \sum_{i=0}^{N-1} S_n^2(i)} \quad \text{--------- (3.1)}$$

Where,
V(n) is the volume,
$S_n(i)$ is the i^{th} sample in the n^{th} frame audio signal,
N is the total number of samples in the frame.
Let us consider three different audio clips Sports, Music, and news for experimental purpose. Figure 3.3 show the waveforms of these clips. The volumes of these audio clips have different distribution.

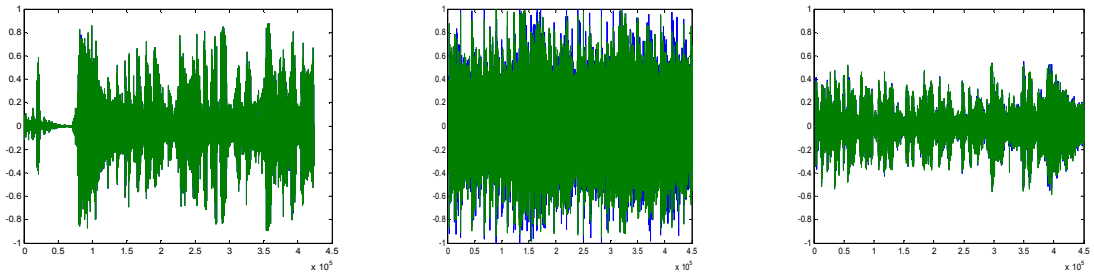


Figure 3.1: Show the Waveform of Different Audio Clip

3.1.1 Volume Standard Deviation (VSD)

The standard deviation of the distribution of volume (of each frame) is calculated, and is treated as VSD. Thus, it is a clip level feature rather than a frame level feature.

$$\sigma = \sqrt{[\sum(A-\bar{A})^2/(n-1)]} \quad \text{--------- (3.2)}$$

Where σ means standard deviation and $\bar{A}$ is the mean. The sport clips has higher values of VSD.

3.1.2 Volume Dynamic Range (VDR)

In audios with action in the background, volume of the frame does not change much, while in non-action audios, there are silent periods between the speeches, and hence VDR is expected to be higher.VDR is calculated as

$$VDR = [(MAX(v) - MIN(v)) / MAX(v)] \quad \text{---------(3.3)}$$

Where MIN(v) and MAX(v) represent the minimum and maximum volume within a clip respectively.

3.1.3 Zero Crossing Rate (ZCR)

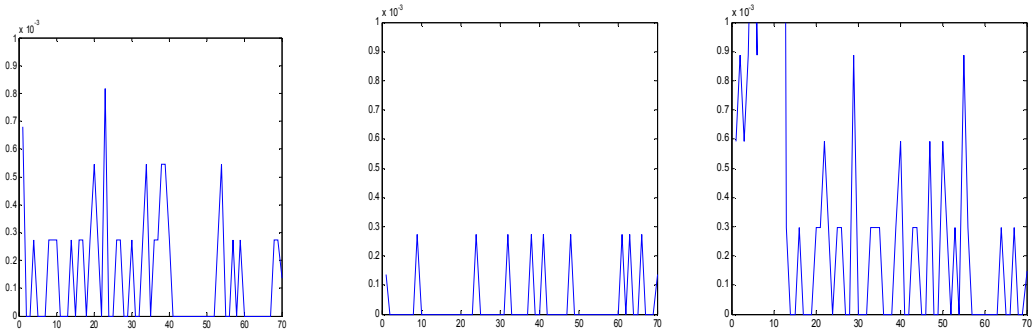
ZCR indicates the number of times that an audio waveform crosses the zero axes. ZCR is the most effective indication for detecting unvoiced speech. By combining ZCR and volume, misclassification of low-energy and unvoiced speech frames as being silent frames can be avoided. Specifically, unvoiced speech is recognized as low volume, but high ZCR [41,76]. Generally, non-action audio samples have a lower ZCR.

The definition of ZCR in discrete case is as follows:

$$ZCR=\frac{1}{2}*(\sum_{i=1}^{N-1}|\text{sgn}[S(n)]-\text{sgn}[S(n-1)]\ |)*\frac{Fs}{N} \quad \text{---------(3.4)}$$

Where,
sgn[S(n)] = 1 S(n) > 0
0 S(n) = 0
-1 S(n) < 0

Where,
S(n) is the input audio signal,
N is the number of signal samples in a frame,
sgn( ) is the sign function and *fs*is the sampling rate of the audio signal.
Figure 3.4 show the ZCR of audio clips. From these plots, we know that the ZCR of these audio clips have different distribution.



(a) Sports (b) Music (c) News

Figure 3.2 Show the ZCR of Different Genres of Audio Clips

3.1.4 Silence Ratio

Volume Root Mean Square and Volume Standard Deviation are helpful to calculate Silence Ratio (SR) of a frame. Silence ratio is not calculated for each frame, but rather, Volume Root Mean Square and Volume Standard Deviation of each frame is used in calculating SR of a clip. Thus, it is a clip level feature rather than being a frame level feature. The silence ratio is calculated as follows:

$$SR(n) = sr/n \quad \text{----------} \ (3.5)$$

Where 'sr' is initially zero and is incremented by one if VRMS is less than the half of VSD in each frame and 'n' is the total number of frames in the clip. In music and sport clips there is always some noise in the background, which results in a low silence ratio. On the other hand silence ratio is much higher in other genres clips.

3.2 PITCH BASED FEATURES

Pitch serves as an important characteristic of an audio for its robust classification. Pitch information helps derive 3 features, which help in a more accurate and better classification, as compared to other features. Average Magnitude Difference Function (AMDF) is used to determine the pitch of a frame, and is defined as:

$$A_m(n) = \frac{\sum_{i=0}^{N-n-1}|\ s_m(i+1)-s_m(i)\ |}{N-n} \quad \text{---------(3.6)}$$

Where,
$A_m(n)$ = AMDF for n^{th} sample in m^{th} frame
N = number of samples in a frame
$s_m(i)$ = i^{th} sample in m^{th} frame of an audio signal

AMDF is calculated for every sample of the frame. The following is the pitch determination algorithm for a given frame:

(i) Find the global minimum value A_{min} (A_{min} is the minimum value of AMDF function for a given frame)
(ii) Find all local minimum points n_i such that $A(n_i) < A_{min} + \delta$ (where δ is determined on the basis of data) .Thus, these values of n_i are the number of corresponding samples, for which AMDF has local minimum values.

(iii) Clearness of each local minimum is checked. Each n_i ,which have been collected in step 2, are checked for clearance i.e. the difference of $A(n_i)$, and average of $A(n_i-4)$,$A(n_i-3)$,$A(n_i-2)$,$A(n_i-1)$, $A(n_i+1)$, $A(n_i+2)$, $A(n_i+3)$, $A(n_i+4)$, if greater than a certain threshold (again decided on the basis of data), the point is said not to be cleared, and if it is not cleared, the local point is removed. Thus, after this step, we only have those sample numbers, for which AMDF function has a cleared local minimum value.

(iv) With the remaining local points (i.e. after step 2 and step 3 have been applied to the set of samples collected), choose the smallest n_i' (where n_i' is the set of points left after application of step 2 and 3 on the given set of points) value as the pitch period.

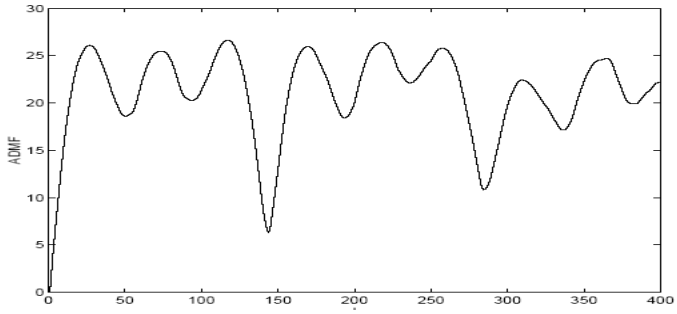


Figure 3.3: Show the AMDF of one News Frame of a audio clip

Thus, the pitch for a frame is calculated by making use of AMDF function for each sample in the frame. Figure 3.5 shows the AMDF of one news clip. Figure 3.6 gives the pitch tracks of the three different audio clips. After computing pitch for each frame, following clip level features are extracted.

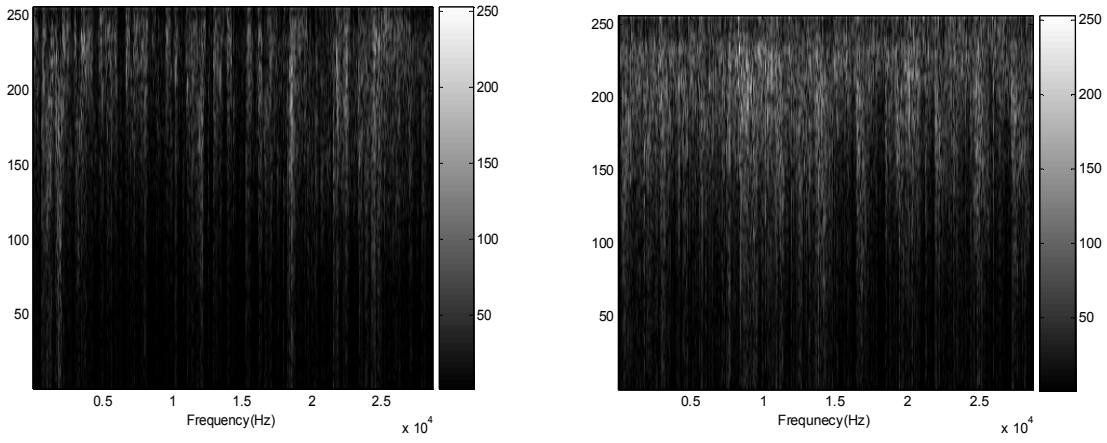
3.2.1 Pitch Standard Deviation (PSD) Pitch standard deviation is calculated using pitch of every frame, and it turns out to be a much better distinguishing factor. We obtained pitch for every frame in a given clip, which helps us to calculate the standard deviation using standard statistics based formulae.

3.2.2 Speech or Music Ratio (VMR) Pitch information of each frame helps to decide whether the frame is speech or music or neither of them. For a speech/music frame, pitch stays relatively smooth. We compare the pitch of a frame with 5 of its neighboring frames, and if all of them are close to each other (decided on the basis of a threshold set on the basis of data provided), then the frame is classified as speech/music. After having gathered information about each frame (i.e. whether it is voice/music or not), we compute VMR as the ratio of these speech/music frames with the total length of the clip:

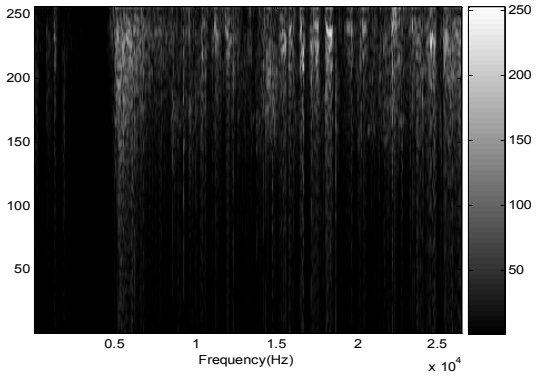
$$VMR = \frac{\text{Number of frames classified as speech/music}}{\text{Total number of frames}} \quad \text{--------(3.7)}$$

3.3 FREQUENCY DOMAIN BASED FEATURES

To obtain frequency domain features, spectrogram of an audio clip, in the form of short-time Fourier transform is calculated for each audio frame. Since time domain does not show the frequency components and frequency distribution of a sound signal [72]. Any sound signal can be expressed in the form of the frequency spectrum which shows the graph of frequency versus amplitude; thus it shows the average amplitude of various frequency components in the audio signal [6]. The spectrogram is used for the extraction of two features, namely frequency Centroid, and frequency bandwidth. Figure 3.7 gives the spectrogram of the three audio clips:



(a) Sports (b) Music



(c) News

Figure 3.4: Show the Frequency Spectrum of Three Different Audio Clips

3.4 ENERGY FEATURES

The energy distribution in different frequency bands also varies quite significantly among different types of audio signals. The entire frequency spectrum is divided into four sub-bands at the same interval of 1 KHz. Each subband consists of six critical bands which represent cochlear filter in the human auditory model [56]. The Sub-band energy is defined as:

$$E_i = \sum_{w=w_{iL}}^{w_{iH}} |F(w)|^2 \quad 1 \le i \le 4 \qquad \text{--------- (3.11)}$$

Here W_{iL} and W_{iH} are lower and upper bound of sub-band i, and Then $E_i$is normalized as:

$$E_i = E_i \Big/ \sum_i E_i \quad 1 \le i \le 4 \qquad \text{--------- (3.12)}$$

3.5 MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCC)

MFCCs are short term spectral based features. MFCC features are frequently used by many researchers for speech recognition and in music/ speech classification problem. A block diagram showing the steps taken for the computing MFFCs can be seen in figure 3.10. Each step in this process of creating Mel Frequency Cepstral Coefficients is motivated by computational or perceptual considerations.

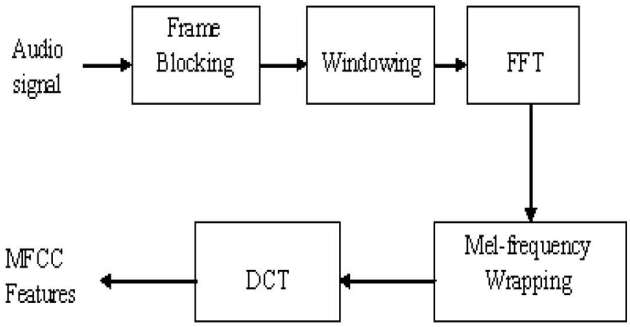


Figure 3.5 Block diagram showing the steps for computing MFCCs

The first step in this process is to block a continuous audio signal into frames. The purpose here is to model small sections of the audio signal that are statistically stationary. Each frame consists of *n* samples with adjacent frames separated by *m* samples. The following frame starts *m* samples after the first sample and overlaps it by *(n - m)* samples. In a similar way the third frame starts *m* samples after the second frame and overlaps it by *(n - m)* samples. Typical values for *n* and *m* are 256 and 100 respectively. The next step is to use a window function on each individual frame in order to minimise discontinuities at the beginning and end of each frame. Typically the window function used is the Hamming window and has the following form:

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N-1}\right), & 0 \le n \le (N-1) \\ 0 \qquad\qquad , otherwise \end{cases} \qquad \text{--------- (3.13)}$$

Given the above window function and assuming that there are *N* samples in each frame, we will obtain the following signal after windowing.

$$y(n) = x(n)w(n) \quad , \quad 0 \le n \le (N-1) \text{--------- (3.14)}$$

The next step is the process of converting each frame of *N* samples from the time domain to the frequency domain. Here we will take the Discrete Fourier Transform of each frame. We use the FFT algorithm, which is computationally efficient, to implement the DFT. As the amplitude of the spectrum is much more important than the phase, we will retain only the amplitude spectrum. The Discrete Fourier Transform on the set of *N* samples is defined as follows [33].

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{\frac{-j2\pi kn}{N}} \quad , \quad k=0,1,2,\ldots\ldots\ldots,(N-1) \qquad \text{--------- (3.15)}$$

The next step is the transformation of the real frequency scale to the mel frequency scale. A mel is a unit of measure of *perceived pitch or frequency* of a tone [45]. The mel- frequency is based on the nonlinear human perception of the frequencies of audio signals. It is a linear frequency spacing below 1KHz and logarithmic above this frequency. By taking the *pitch* of the 1 KHz tone as a reference and assigning it 1000 mels, and then making some test measurements based on human perception of audio signals, it is possible to drive a model for an approximate mapping of a given real frequency to the mel frequency scale. The following is an approximation of the mel frequency based on such experiments.

$$Mel(f)=2595\log_{10}(1+\frac{f}{700}), \qquad \text{--------- (3.16)}$$

Where *f* is the physical frequency in *Hz* and *Mel* is the perceived frequency in *mels*.

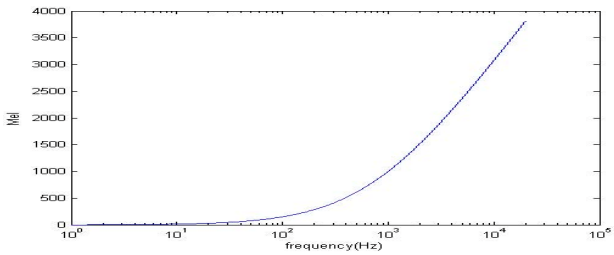


Figure 3.6 Plot of a news signal as function of time

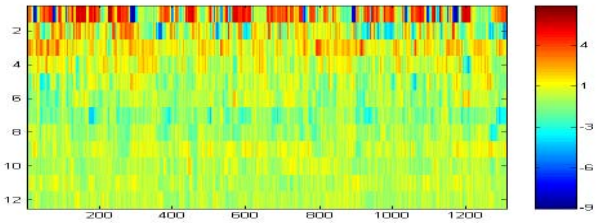


Figure 3.7 Plot of the MFCCs for the News signal

3.6 ENTROPY

Entropy is a property that can be used to determine the energy not available for work. It is also a measure of the tendency of a process. It is a measure of disorder of a system. Entropy refers to the relative degree of randomness. The higher the entropy, the more frequently are signaling errors. Entropy is directly proportional to the maximum attainable data speed in bps. Entropy is directly proportional to noise and bandwidth. It is inversely proportional to compatibility. Entropy also refers to disorder deliberately added to data in certain encryption process.

IV. NEURAL NET BASED AUDIO CLASSIFICATION

Recent awareness in artificial neural networks has motivated a large number of applications covering a wide range of research fields. The ability of learning in neural networks provides an interesting alternative to other conventional research methods. The different problem domains where neural network may be used are: pattern association, pattern classification, regularity detection, image processing, speech analysis, simulation etc. We have designed a neural classifier using artificial neural networks for our Content based audio classification task. In this chapter, we first describe the neural network with its various types, learning methodology of neural network and designing of neural network classifier.

4.1 NEURAL NETWORK

A neural network is an artificial representation of the human brain that tries to simulate its learning process. The term "artificial" means that neural nets are implemented in computer programs that are able to handle the large number of necessary calculations during the learning process [1, 10, 39].The human brain consists of a large number (more than a billion) of neural cells that process information. Each cell works like a simple processor and only the massive interaction between all cells and their parallel processing makes the brain's abilities possible.

Figure 4.1 shows a sketch of such a neural cell, called a neuron. In figure, a neuron consists of a core, dendrites for incoming information and an axon with dendrites for outgoing information that is passed to connected neurons. Information is transported between neurons in form of electrical stimulations along the dendrites. Incoming information's that reach the neuron's dendrites is added up and then delivered along the neuron's axon to the dendrites at its end, where the information is passed to other neurons if the stimulation has exceeded a certain threshold the neuron is said to be activated. If the incoming stimulation has been too low, the information will not be transported any further and the neuron is said to be inhibited. Like the human brain, a neural net also consists of neurons and connections between them. The neurons are transporting incoming information on their outgoing connections to other neurons. In neural net terms these connections are called weights. The "electrical" information is simulated with specific values stored in these weights. By simply changing these weight values the changing of the connection structure can also be simulated. Figure 4.2 shows an idealized neuron of a neural net. In this network, information (called the input) is sent to the neuron on its incoming weights. This input is processed by a propagation function that adds up the values of all incoming weights. The resulting value is compared with a certain threshold value by the neuron's activation function. If the input exceeds the threshold value, the neuron will be activated, otherwise it will be inhibited. If activated, the neuron sends an output on its outgoing weights to all connected neurons and so on.

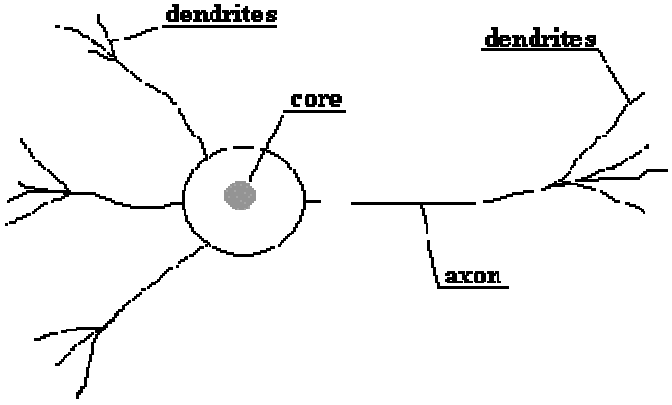


Figure 4.1: Structure of a neural cell in the human brain

In a neural net, the neurons are grouped in layers, called neuron layers. Usually each neuron of one layer is connected to all neurons of the preceding and the following layer (except the input layer and the output layer of the net). The information given to a neural net is propagated layer-by-layer from input layer to output layer through none, one or more hidden layers. Depending on the learning algorithm, it is also possible that information is propagated backwards through the net. Figure 4.3 shows a neural net with three neuron layers. This is not the general structure of a neural net. Some neural net types have no hidden layers or the neurons in a layer are arranged as a matrix. But the common to all neural net types is the presence of at least one weight matrix and the connections between two neuron layers.

4.2.1 Perceptron

The perceptron was first introduced by F. Rosenblatt in 1958. It is a very simple neural net type with two neuron layers that accepts only binary input and output values (0 or 1). The learning process is supervised and the net is used for pattern classification purposes. Figure 4.4 shows the simple Perceptron:

4.2.2 Multi Layer Perceptron

The Multi-Layer-Perceptron was first introduced by M. Minsky and S. Papert in 1969. It is an extended Perceptron and has one ore more hidden neuron layers between its input and output layers as shown in figure 4.5.

4.2.3 Back Propagation Network

The Back propagation Network was first introduced by G.E. Hinton, E. Rumelhart and R.J. Williams in 1986 and is one of the most powerful neural net types. It has the same structure as the Multi-Layer-Perceptron and uses the back propagation learning.

4.3 LEARNING

In the human brain, information is passed between the neurons in form of electrical stimulation along the dendrites. If a certain amount of stimulation is received by a neuron, it generates an output to all other connected neurons and so information takes its way to its destination where some reaction will occur. If the incoming stimulation is too low, no output is generated by the neuron and the information's further transport will be blocked. Explaining how the human brain learns certain things is quite difficult and nobody knows it exactly. It is supposed that during the learning process the connection structure among the neurons is changed, so that certain stimulations are only accepted by certain neurons. This means, there exist firm connections between the neural cells that once have learned a specific fact, enabling the fast recall of this information.

If some related information is acquired later, the same neural cells are stimulated and will adapt their connection structure according to this new information. On the other hand, if specific information isn't recalled for a long time, the established connection structure between the responsible neural cells will get more "weak". This has happened if someone "forgot" a once learned fact or can only remember it vaguely. Unlike the biological model, a neural net has an unchangeable structure, built of a specified number of neurons and a specified number of connections between them (called "weights"), which have certain values. What changes during the learning process are the values of those weights? Compared to the original this means: Incoming information "stimulates" (exceeds a specified threshold value of) certain neurons that pass the information to connected neurons or prevent further transportation along the weighted connections. The value of a weight will be increased if information should be transported and decreased if not. While learning different inputs, the weight values are changed dynamically until their values are balanced, so each input will lead to the desired output. The training of neural net results in a matrix that holds the weight values between the neurons. Once a neural net has been trained correctly, it will probably be able to find the desired output to a given input that has been learned, by using these matrix values. Very often there is a certain error left after the learning process, so the generated output is only a good approximation to the perfect output in most cases.

4.3.2 Back Propagation Learning Algorithm

Back propagation is a supervised learning algorithm and is mainly used by Multi Layer Perceptron to change the weights connected to the net's hidden neuron layer(s). The back propagation algorithm uses a computed output error to change the weight values in backward direction. To get this net error, a forward propagation phase must have been done before. While propagating in forward direction, the neurons are being activated using the sigmoid activation function.

Following are the steps of back propagation algorithm:

(i)Initialize the weights and biases:

- The weights in the network are initialized to random numbers from the interval [-1,1].
- Each unit has a BIAS associated with it
- The biases are similarly initialized to random numbers from the interval [-1,1].

(ii) Feed the training sample:

Each training sample is processed by the steps given below.

(iii) Propagate the inputs forward:

We compute the net input and output of each unit in the hidden and output layers.

V. IMPLEMENTATION AND EXPERIMENTAL RESULTS

Overall schematic diagram for audio classifier system is shown in figure 5.1. Input to the system is an audio clip, while the output is a genre of the defined types. All the work is undertaken using WAV audio clips.

5.1 IMPLEMENTATION OF AUDIO CLASSIFIER

The audio classifier is implemented using MATLAB Artificial Neural Network toolbox in two phases. In the first phase the audio is classified into three classes' sports, news and music audio. In the second phase the music clips are classified as angry, sad and happy mood.

5.1.1 Implementation of Audio Classifier in First Phase

Following are the steps of implementation of Audio classification system for first phase:

5.1.1.1 Data Preparation

We have extracted the 14 audio features of 122 audio clips of three different genres and store them in a file **Audiodata.m** The extracted audio features data is normalized (i.e. all values of attributes in the database are changed to contain values in the internal [0,1]) using Max-Min normalization technique. The normalized data is dividing them into training data and testing data. Training and testing data sets are usually split in a roughly 3:1 ratio. This is done with the following commands:

- Load the file into MATLAB with the following command:
  Audio = load('Audiodata.m');
- We extract the training data into a matrix called TrainSet using.
  TrainSet = [Audio(1:40,:); Audio(51:74,:); Audio(83:113,:)];
- Since we know which rows correspond to which class, we can simply dump the final column of the data:
  TrainSet = TrainSet(:,1:20);
- This is repeated for the testing data set with the following commands:
  TestSet = [Audio(41:50,:); Audio(75:82,:); Audio(114:124,:)];
  TestSet = TestSet(:,1:20);
- We then need to create the output examples for the training set. This is done as follows. Firstly, create a matrix of zeros, as follows:
  TrainOut = zeros (94, 3);

- Then, set the first column of the first 40 rows to one which represents the first class
  TrainOut(1:40,1) = 1;
- Similarly, Set the second column of the 24 rows(41-64), third column of the 30 rows(65-94), to one respectively using the commands:
  TrainOut(41:64,2) =1;
  TrainOut(65:94,3) =1;
- This will associate the output vector 1 0 0 with the first class (sports), 0 1 0 with the second class (music), and 0 0 1 with the third class (news).

5.1.1.2 Creating a Multi-Layer Feed Forward Network

With the prepared data sets, we now create the multi layer feed forward network. This is done with the **newff** function, which is part of the Neural Network Toolbox. Function has four arguments:

- **inp** is the range of the input values, that can be determined using the minmax function, as follows:
  Audio = Audio (:, 1:20);
  Audio=Audio';
  inp = minmax(Audio);
- **neuron_layer** is a one row matrix specifying the number of neurons in the hidden and output neuron layers. The number of input neurons is inferred from the number of rows in the **inp** matrix. We have five output neurons (one for each class) and ten hidden neurons. These parameters are encoded into the matrix as follows:
  neuron_layer = [10 3];
- It is also necessary to specify the activation functions for each neuron layer. We used a sigmoid function **logsig** for the hidden and output neuron layers which are specified in an array of strings (called a 'cell array'), as follows:
  func = {'logsig', 'logsig'};
- The final parameter is the training method to be used on the network, which is specified as **traingd**.
- Finally the network is created by the following command:
  AudioNet = **newff**(inp,neuron_layer,func,'traingd');

5.1.1.3 Experimental Results of First Phase Classification

We have conducted extensive experiments on around two hundred audio clips. These clips were obtained from the website. Our experiments show interesting structure within the feature space, implying that a mapping does indeed exist between high-level classification and low-level computable features. We identified three major classes, namely sports, news, and music. The easiest way of examining the accuracy of the trained classifier is by plotting the output value of each output neuron for each audio clip. For the first ten audio clips, the first output neuron should be the most highly activated, for the second eight audio clips, the second output neuron should be activated and so on. Figure 5.12 shows the graph of the output values for nine audio clips. These values are generated by the audio classifier. The following commands are used to draw the graph:

```
out1 = out(1,:);
out2 = out(2,:);
out3 = out(3,:);
plot(out1,'+');
hold;
plot(out2,' o');
plot(out3,'d');
```

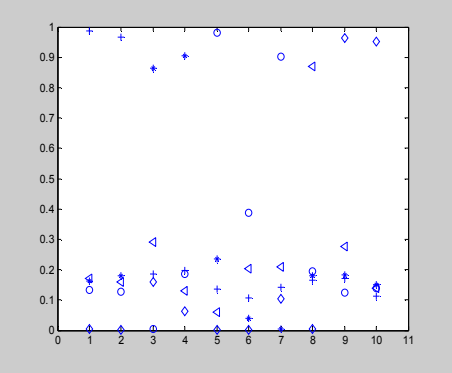


Figure 5.1 Graph of the Output Values of Neurons for Ten audio Clips

In the graph, X axis represents the clip serial no. and Y axis represents the classification values of the neurons. We have used the following symbols for representation of movie genres.
+ (plus) – Sports
O (circle) – News
D (diamond) – Musical
The table 5.7 shows the output values of neurons for nine audio clips generated by the audio classifier. The output neuron1 represent the sports, neuron2 represent the news and so on.

Audio Classifier in Second Phase

Further music is classified into angry, sad and happy mood on the basis of same set of low level audio features. Following are the steps of implementation of Audio classification system in second phase:

VI. CONCLUSIONS AND FUTURE SCOPE

We have attempted to combine various features extracted from audio clips for segmenting and characterizing it into different categories of audio. We have elaborated our work to the characterization of the audio using low level audio features. For this we have extracted audio features like Time domain based, pitch based, frequency based, sub band energy based and MFCC. We have designed audio classifier using multi layer feed forward neural network with supervised back propagation learning algorithm. The network is trained extracted audio features. The system could classify the audio clip in to sports, News, and music. Further the system is also used to classify the music in to mood such as angry, sad and happy. The results obtained for the classification of audio are quite promising. This has been in-spite of the fact that our training set has been fairly small and consists of audio clips of different semantic structures. This authenticates the choice of features that we have selected. We have trained our system for a fairly small database and the length of the audio clips is also small (60 seconds). An immediate concern would be to see how it scales to huge databases and full length of audios for it to have any commercial value. In the present system, we are only able to characterize audio in to sports, News, and music. In Future work, we could further extend the work by including more categories of audio for characterization in the existing system. The efficiency of our system to classify audio is around 80%. But still there are great chances to improve the efficiency of the system.

REFERENCES

1. C.C. Liu, Y.H. Yang, P.H. Wu, and H.H. Chen.Detecting and classifying emotion in popular music. In JCIS, 2006.
2. C.H. Chen, M.F. Weng, S.K. Jeng, and Y.Y. Chuang.Emotion-Based Music Visualization Using Photos.LNCS, 4903:358–368, 2008
3. C. McKay and I. Fujinaga, “Automatic Genre Classification Using Large High-Level Musical Feature Sets,” in Proc. of the Int. Conf. on Music Information Retrieval 2004, Barcelona, Spain, Oct. 2004.
4. D. Manoussaki, R.S. Chadwick, D.R. Ketten, J. Arruda, E. Dimitriadis, and J.T. O’Malley. The influence of cochlear shape on low-frequency hearing. ProcNatlAcadSci USA, 105(16):6162–6166, 2008.
5. D. Manoussaki, E. Dimitriadis, and R.S. Chadwick. The cochlea’s graded curvature effect on low frequency waves. Physical Review Letters, 96(8), 2006.
6. D.YangandW.Lee. Disambiguating music emotion using software agents. In Proceedings of the 5th International Conference on Music Information Retrieval, Barcelona,Spain, 2004.
7. E. Didiot, I. Illina, and D. Fohr et al., “A wavelet-based parameterization for speech/music discrimination,” Comput. Speech Lang., vol. 24, no. 2, pp. 341-357, Apr. 2010.
8. E. Pampalk, A. Flexer, and G. Widmer, “Improvements of Audio-Based Music Similarity and Genre Classification,” in Proc. of the 6th Int. Conference on Digital Audio Effects, London, UK, Sept. 2003.
9. J. E. Munoz-Exposito, S. Garcia-Galán, and N. Ruiz-Reyes et al., “Speech/music discrimination using a single warped LPC-based feature,” in Proc. Int. Conf. Music Information Retrieval ’05, London, 2005, pp. 614-617.
10. J. Hao, L. Tong, and Z. Hong-Jiang, "Video segmentation with the assistance of audio content analysis," Multimedia and Expo, 2000. ICME 2000. 2000IEEE International Conference on, vol. 3, pp. 1507-1510 vol.3, 2000.
11. J. J. Burred and A. Lerch, “A Hierarchal Approach to Automatic Musical Genre Classification,” in Proc. of the 6th Int. Conference on Digital Audio Effects, London, UK, Sept. 2003.
12. J. Pinquier, J. Rouas, and R. André-Obrecht, “A fusion study in speech/music classification,” in Proc. Int. Conf. Acoustics, Speech, Signal Processing ’03, Hong Kong, 2003, pp. 409-412.
13. J. Razik, C. Sénac, and D. Fohr et al., “Comparison of two speech/music segmentation systems for audio indexing on the web,” in Proc. Multi Conf. Systemics, Cybernetics, Informatics, Orlando, FL 2003.
14. L. Atlas and S.A. Shamma. Joint acoustic and modulation frequency. EURASIP Journal on Applied Signal Processing, 7:668–675, 2003.
15. L. Lu, D. Liu, and H.J. Zhang. Automatic mood detection and tracking of music audio signals. IEEE Trans. Audio, Speech & Language Process, 14(1), 2006.
16. L. Lu, H. Jiang, and H. Zhang, "A robust audio classification and segmentation method," presented at Proceedings of the ninth ACM international conference on Multimedia, Ottawa, Canada, 2001.
17. Sanjay Jain and R.S. Jadon, “Audio Based Movies Characterization using Neural Network”, published in International Journal of Computer Science and Applications(IJCSA ISSN 0974-1003), Vol 1 No.2, PP 87- 91, Aug, 2008.
18. Sanjay Jain and R.S. Jadon, “Features Extraction for Movie Genres Characterization”, in Proceeding of WCVGIP-06, 2006.
19. S. Kim, S. Kim, S. Kwon, and H. Kim. A music summarization scheme using tempo racking and two stage clustering. IEEE Workshop on Multimedia Signal Processing, pages 225–28,2006.

