RIEA – Retrieval of Information & Emotion Analysis of Music

A thesis submitted in partial fulfillment of the requirement for the degree of Master of Technology in Computer Technology in the Department of Computer Science & Engineering

Of

Jadavpur University

By

Oindrila Dutta

Registration No: 133935 0f 2015-2016

Examination Roll No.: M6TCT22050B

Under the Guidance of

Dr Dipankar Das

Department of Computer Science & Engineering

Jadavpur University, Kolkata-700032

2022

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

To Whom It May Concern

I hereby recommend that the thesis entitled "**RIEA – Retrieval of Information & Emotion Analysis of Music**" has been carried out by Oindrila Dutta (Reg. No.: 133935 of 2015-2016, Exam Roll: M6TCT22050B), under my guidance and supervision and be accepted in partial fulfillment

of the requirement for the degree of Master of Technology in Computer Technology in the Department of Computer Science and Engineering, Jadavpur University.

…………………………………

Prof.(Dr) Nandini Mukherjee
Head of Department
Department of Computer Science
and Engineering
Jadavpur University,
Kolkata-700032

…………………………………

Dr Dipankar Das
Thesis Supervisor
Department of Computer Science
and Engineering
Jadavpur University,
Kolkata-700032

…………………………………

Dean
Dr Bhaskar Gupta
Faculty of Engineering and
Technology
Jadavpur University,
Kolkata-700032

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Certificate of Approval

This is to certify that the thesis entitled "**RIEA – Retrieval of Information & Emotion Analysis of Music**" is a bonafide record of work carried out by Oindrila Dutta in fulfillment of the requirements for the award of the degree of Master of Technology in Computer Technology in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approves the thesis only for the purpose for which it has been submitted.

…………………………………………………………………………

Signature of Examiner 1

Date:

…………………………………………………………………………

Signature of Examiner 2

Date:

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Declaration of Originality & Compliance of Academic Ethics

I hereby declare that this thesis contains a literature survey and original research work by the undersigned candidate, as a part of his Master of Technology in Computer Technology studies.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Oindrila Dutta

Registration No: 133935 of 2015-2016

Exam Roll No.: M6TCT22050B

Thesis Title: RIEA – Retrieval of Information & Emotion Analysis of Music

……………………………………………….

Signature with Date

Acknowledgement

The writing of the thesis and the related work has been a long journey with input from many individuals, right from the first day till the development of the final project.

With my most sincere gratitude, I would like to thank Dr Dipankar Das, my supervisor, for his overwhelming support throughout the duration of the project. His motivation always gave me the required input and momentum to continue with my work, without which the project work would not have taken its current shape. His valuable suggestions and numerous discussions have always inspired new ways of thinking. I feel deeply honored that I got this opportunity to work under him.

I would like to thank all the faculty members of the Department of Computer Science and Engineering of Jadavpur University for their continuous support.

Last, but not least, I would like to thank all my batch mates of Master of Technology in Computer Technology, Jadavpur University for staying by my side when I needed them.

………………………………………………………

Name: Oindrila Dutta
Examination Roll No.: M6TCT22050B
University Registration No: 133935 of 2015-2016

ABSTRACT

Sentiment analysis has evolved over the past few decades, most of the work in it revolved around textual sentiment analysis with text mining techniques. But audio sentiment analysis is still in a nascent stage in the research community. In this proposed research, we perform sentiment analysis on speaker-discriminated speech transcripts to detect the emotions of the individual speakers involved in the conversation. We analysed different techniques to perform speaker discrimination and sentiment analysis to find efficient algorithms to perform this task.

TABLE OF CONTENTS

INTRODUCTION

1.1 **Music Information Retrieval**

Music information retrieval (MIR) is the interdisciplinary science of retrieving information from music. MIR is a small but growing field of research with many real-world applications. Those involved in MIR may have a background in musicology, psychoacoustics, psychology, academic music study, signal processing, informatics, machine learning, optical music recognition, computational intelligence or some combination of these.

1.2 **Types of Analysis**

1.2.1 **Sentimental Analysis**

Sentiment analysis (also known as opinion mining or emotion AI) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

Sentiment analysis is the automated process that uses AI to identify positive, negative and neutral opinions from the text. Sentiment analysis is widely used for getting insights from social media comments, survey responses, and product reviews, and making data-driven decisions.

In a world where we generate 2.5 quintillion bytes of data every day, sentiment analysis has become a key tool for making sense of that data.

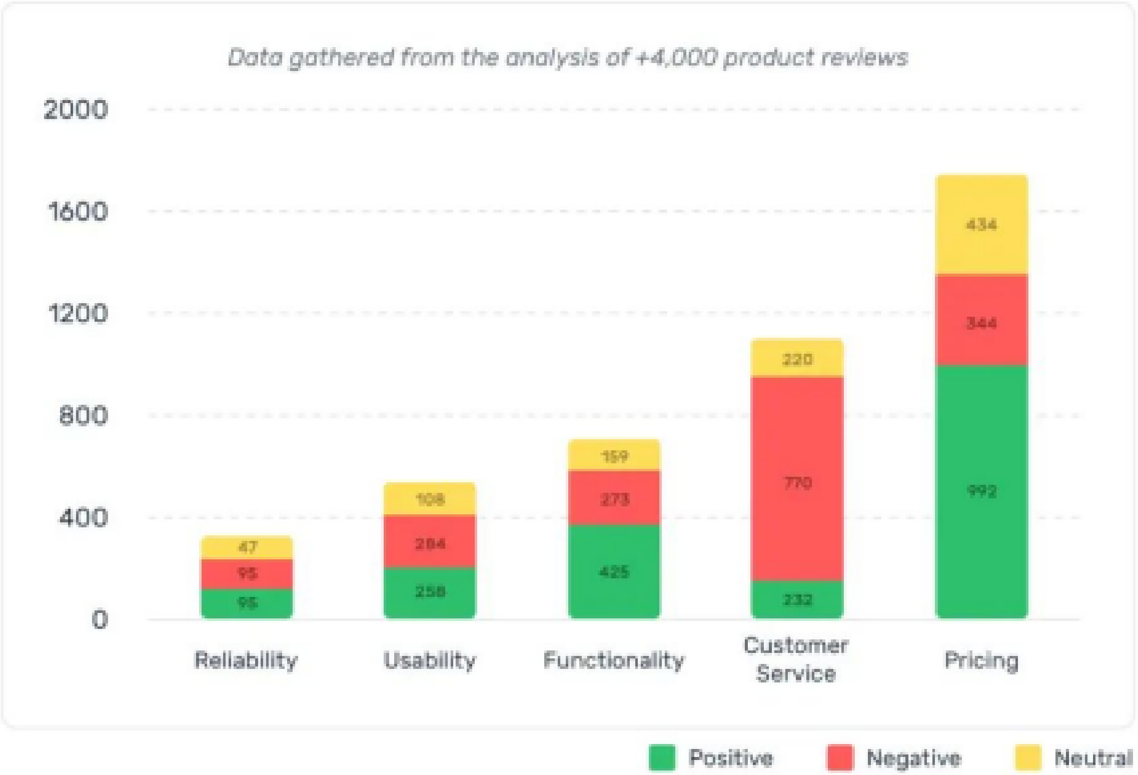
Sentiment analysis is the automated process of analyzing text data and classifying opinions as negative, positive or neutral. Usually, besides identifying the opinion, these systems extract attributes of the expression e.g.:

Polarity: if the speaker expresses a positive or negative opinion, **Subject**: the thing that is being talked about,

Opinion holder: the person, or entity that expresses the opinion.

Currently, sentiment analysis is a topic of great interest and development since it has many practical applications. Companies use sentiment analysis to automatically analyze survey responses, product reviews, social media comments, and the like to get valuable insights about their brands, product, and services.

For example, one of our customers used sentiment analysis to automatically analyze 4,000+ reviews and better understand how their customers perceived their product. They found out that customers were generally happy about the pricing but complaining a lot about their customer service:



There are many types and flavors of sentiment analysis and SA tools range from systems that focus on polarity (positive, negative, neutral) to systems that detect feelings and emotions (angry, happy, sad, etc) or identify intentions (e.g. interested v. not interested). In the following section, we'll cover the most important ones.

1.2.1.1 **Fine-grained Sentiment Analysis**

Sometimes you may be also interested in being more precise about the level of the polarity of the opinion, so instead of just talking about positive, neutral, or negative opinions you could consider the following categories:

- Very positive
- Positive
- Neutral
- Negative
- Very negative

This is usually referred to as fine-grained sentiment analysis. This could be, for example, mapped onto a 5-star rating in a review,

e.g.: Very Positive = 5 stars and Very Negative = 1 star.

Some systems also provide different flavors of polarity by identifying if the positive or negative sentiment is associated with a particular feeling, such as anger, sadness, or worries (i.e. negative feelings) or happiness, love, or enthusiasm (i.e. positive feelings).

1.2.1.2 **Emotion detection**

Emotion detection aims at detecting emotions like happiness, frustration, anger, sadness, and the like. Many emotion detection systems resort to lexicons (i.e. lists of words and the emotions they convey) or complex machine learning algorithms.

One of the downsides of resorting to lexicons is that the way people express their emotions varies a lot and so do the lexical items they use. Some words that would typically express anger like shit or kill (e.g. in your product is a piece of shit or your customer support is killing me) might also express happiness (e.g. in texts like This is the shit or You are killing it).

1.2.1.3 **Aspect-based Sentiment Analysis**

Usually, when analyzing the sentiment in subjects, for example, products, you might be interested in not only whether people are talking with a positive, neutral, or negative polarity about the product, but also which particular aspects or features of the product people talk about. That's what aspect-based sentiment analysis is about. In our previous example:

"*The battery life of this camera is too short.*"

The sentence is expressing a negative opinion about the camera, but more precisely, about the battery life, which is a particular feature of the camera.

1.2.1.4 **Intent analysis**

The intent analysis basically detects what people want to do with a text rather than what people say with that text. Look at the following examples:

“Your customer support is a disaster. I’ve been on hold for 20 minutes”.

“I would like to know how to replace the cartridge”.

“Can you help me fill out this form?”

A human being has no problems detecting the complaint in the first text, the question in the second text, and the request in the third text. However, machines can have some problems to identify those. Sometimes, the intended action can be inferred from the text, but sometimes, inferring it requires some contextual knowledge.

1.2.1.5 Multilingual sentiment analysis

Multilingual sentiment analysis can be a difficult task. Usually, a lot of pre-processing is needed and that pre-processing makes use of a number of resources. Most of these resources are available online (e.g. sentiment lexicons), but many others have to be created (e.g. translated corpora or noise detection algorithms). The use of the resources available requires a lot of coding experience and can take longer to implement.

An alternative to that would be detecting the language in texts automatically, then train a custom model for the language of your choice (if texts are not written in English), and finally, perform the analysis.

1.2.2 Importance of Sentiment Analysis

It's estimated that 80% of the world's data is unstructured and not organized in a pre-defined manner. Most of this comes from text data, like emails, support tickets, chats, social media, surveys, articles, and documents. These texts are usually difficult, time-consuming and expensive to analyze, understand, and sort through.

Sentiment analysis systems allow companies to make sense of this sea of unstructured text by automating business processes, getting actionable insights, and saving hours of manual data processing, in other words, by making teams more efficient.

1.2.3 Multimodal Sentiment Analysis

Multimodal sentiment analysis is a new dimension of the traditional text-based sentiment analysis, which goes beyond the analysis of texts, and includes other modalities such as audio and visual data. It can be bimodal, which includes different combinations of two modalities, or trimodal, which incorporates three modalities. With the extensive amount of social media data available online in different forms such as videos and images, the conventional text-based sentiment analysis has evolved into more complex models of multimodal sentiment analysis, which can be applied in the development of virtual assistants, analysis of YouTube movie reviews, analysis of news videos, and emotion recognition (sometimes known as emotion detection) such as depression monitoring, among others.

Similar to the traditional sentiment analysis, one of the most basic tasks in multimodal sentiment analysis is sentiment classification, which classifies different sentiments into categories such as positive, negative, or neutral. The complexity of analyzing text, audio, and visual features to perform such a task requires the application of different fusion techniques, such as feature-level, decision-level, and hybrid fusion. The performance of these fusion techniques and the classification algorithms applied, are influenced by the type of textual, audio, and visual features employed in the analysis.

1.2.3.1 **Features**

Feature engineering, which involves the selection of features that are fed into machine learning algorithms, plays a key role in the sentiment classification performance. In multimodal sentiment analysis, a combination of different textual, audio, and visual features are employed.

1.2.3.2 **Textual features**

Similar to the conventional text-based sentiment analysis, some of the most commonly used textual features in multimodal sentiment analysis are unigrams and n-grams, which are basically a sequence of words in a given textual document. These features are applied using bag-of-words or bag-of-concepts feature representations, in which words or concepts are represented as vectors in a suitable space.

1.2.3.3 **Audio features**

Sentiment and emotion characteristics are prominent in different phonetic and prosodic properties contained in audio features. Some of the most important audio features employed in multimodal sentiment analysis are Mel-frequency cepstrum (MFCC), spectral centroid, spectral flux, beat histogram, beat sum, strongest beat, pause duration, and pitch. Praat is a popular open-source toolkit for extracting such audio features.

1.2.3.4 **Visual features**

One of the main advantages of analyzing videos with respect to texts alone is the presence of rich sentiment cues in visual data. Visual features include facial expressions, which are of paramount importance in capturing sentiments and emotions, as they are the main channel of forming a person's present state of mind. Specifically, a smile, is considered to be one of the most predictive visual cues in multimodal sentiment analysis. OpenFace is an open-source facial analysis toolkit available for extracting and understanding such visual features.

1.2.3.5 Fusion techniques

Unlike the traditional text-based sentiment analysis, multimodal sentiment analysis undergoes a fusion process in which data from different modalities (text, audio, or visual) are fused and analyzed together. The existing approaches in multimodal sentiment analysis data fusion can be grouped into three main categories: feature-level, decision-level, and hybrid fusion, and the performance of the sentiment classification depend on which type of fusion technique is employed.

1.2.3.6 Feature-level fusion

Feature-level fusion (sometimes known as early fusion) gathers all the features from each modality (text, audio, or visual) and joins them together into a single feature vector, which is eventually fed into a classification algorithm. One of the difficulties in implementing this technique is the integration of heterogeneous features.

1.2.3.7 Decision-level fusion

Decision-level fusion (sometimes known as late fusion), feeds data from each modality (text, audio, or visual) independently into its own classification algorithm, and obtains the final sentiment classification results by fusing each result into a single decision vector. One of the advantages of this fusion technique is that it eliminates the need to fuse heterogeneous data, and each modality can utilize its most appropriate classification algorithm.

1.2.3.8 Hybrid fusion

Hybrid fusion is a combination of feature-level and decision-level fusion techniques, which exploit complementary information from both methods during the classification process. It usually involves a two-step procedure wherein feature-level fusion is initially performed between two modalities, and decision-level fusion is then applied as a second step, to fuse the initial results from the feature-level fusion, with the remaining modality.

1.3 Advantages of analysis

1.3.1 Scalability

Can you imagine manually sorting through thousands of tweets, customer support conversations, or customer reviews? There's just too much data to process manually. Sentiment analysis allows processing data at scale in an efficient and cost-effective way.

1.3.2 Real-time analysis

We can use sentiment analysis to identify critical information that allows situational awareness during specific scenarios in real-time. Is there a PR crisis in social media about to burst? An angry customer is about to churn? A sentiment analysis system can help you immediately identify these kinds of situations and take action.

1.3.3 Consistent criteria

Humans don't observe clear criteria for evaluating the sentiment of a piece of text. It's estimated that different people only agree around 60-65% of the time when judging the sentiment for a particular piece of text. It's a subjective task that is heavily influenced by personal experiences, thoughts, and beliefs. By using a centralized sentiment analysis system, companies can apply the same criteria to all of their data. This helps to reduce errors and improve data consistency.

Check out the Use Cases & Applications section to see examples of companies and organizations that are using sentiment analysis for a diverse set of things.

1.4 Analysis Algorithms

There are many methods and algorithms to implement sentiment analysis systems, which can be classified as:

- Rule-based systems that perform sentiment analysis based on a set of manually crafted rules.
- Automatic systems that rely on machine learning techniques to learn from data.
- Hybrid systems that combine both rule-based and automatic approaches.

1.4.1 **Rule-based Approaches**

Usually, rule-based approaches define a set of rules in some kind of scripting language that identify subjectivity, polarity, or the subject of opinion.

The rules may use a variety of inputs, such as the following:

- Classic NLP techniques like stemming, tokenization, part of speech tagging and parsing.
- Other resources, such as lexicons (i.e. lists of words and expressions).

A basic example of a rule-based implementation would be the following:

Define two lists of polarized words (e.g. negative words such as bad, worst, ugly, etc and positive words such as good, best, beautiful, etc).

Given a text:
Count the number of positive words that appear in the text.
Count the number of negative words that appear in the text.

If the number of positive word appearances is greater than the number of negative word appearances return a positive sentiment, conversely, return a negative sentiment. Otherwise, return neutral.

This system is very naïve since it doesn't take into account how words are combined in a sequence. More advanced processing can be made, but these systems get very complex quickly. They can be very hard to maintain as new rules may be needed to add support for new expressions and vocabulary. Besides, adding new rules may have undesired outcomes as a result of the interaction with previous rules. As a result, these systems require important investments in manually tuning and maintaining the rules.

1.4.2 **Automatic Approaches**

Automatic methods, contrary to rule-based systems, don't rely on manually crafted rules, but on machine learning techniques. The sentiment analysis task is usually modeled as a classification problem where a classifier is fed with a text and returns the corresponding category, e.g. positive, negative, or neutral (in case of polarity analysis is being performed).

1.4.3 **Classification Algorithms**

The classification step usually involves a statistical model like Naïve Bayes, Logistic Regression, Support Vector Machines, or Neural Networks:

1.4.4 **Naïve Bayes**

A family of probabilistic algorithms that uses Bayes's Theorem to predict the category of a text.

1.4.5 **Linear Regression**

A very well-known algorithm in statistics used to predict some value (Y) given a set of features (X).

1.4.6 **Support Vector Machines**

A non-probabilistic model that uses a representation of text examples as points in a multidimensional space. These examples are mapped so that the examples of the different categories (sentiments) belong to distinct regions of that space. Then, new texts are mapped onto that same space and predicted to belong to a category based on which region they fall into.

1.4.7 **Deep Learning**

A diverse set of algorithms that attempts to imitate how the human brain works by employing artificial neural networks to process data.

1.5 **Applications of MIR**

MIR is being used by businesses and academics to categorize, manipulate and even create music.

1.5.1 **Recommender systems**

Several recommender systems for music already exist, but surprisingly few are based upon MIR techniques, instead of making use of similarity between users or laborious data compilation. Pandora, for example, uses experts to tag the music with particular qualities such as "female singer" or "strong bassline". Many other systems find users whose listening history is similar and suggests unheard music to the users from their respective collections. MIR techniques for similarity in music are now beginning to form part of such systems.

1.5.2 **Track separation and instrument recognition**

Track separation is about extracting the original tracks as recorded, which could have more than one instrument played per track. Instrument recognition is about identifying the instruments involved and/or separating the music into one track per instrument. Various programs have been developed that can separate music into its component tracks without access to the master copy. In this way e.g. karaoke tracks can be created from normal music tracks, though the process is not yet perfect owing to vocals occupying some of the same frequency space as the other instruments.

1.5.3 **Automatic music transcription**

Automatic music transcription is the process of converting an audio recording into symbolic notation, such as a score or a MIDI file. This process involves several audio analysis tasks, which may include multi-pitch detection, onset detection, duration estimation, instrument identification, and the extraction of harmonic, rhythmic or melodic information. This task becomes more difficult with greater numbers of instruments and a greater polyphony level.

1.5.4 **Automatic categorization**

Musical genre categorization is a common task for MIR and is the usual task for the yearly Music Information Retrieval Evaluation eXchange(MIREX). Machine learning techniques such as Support Vector Machines tend to perform well, despite the somewhat subjective nature of the classification. Other potential classifications include identifying the artist, the place of origin or the

mood of the piece. Where the output is expected to be a number rather than a class, regression analysis is required.

1.5.5 **Music generation**

The automatic generation of music is a goal held by many MIR researchers. Attempts have been made with limited success in terms of human appreciation of the results.

1.5.6 **Data source**

Scores give a clear and logical description of music from which to work, but access to sheet music, whether digital or otherwise, is often impractical. MIDI music has also been used for similar reasons, but some data is lost in the conversion to MIDI from any other format unless the music was written with the MIDI standards in mind, which is rare. Digital audio formats such as “.WAV”, “.mp3”, and “.ogg” are used when the audio itself is part of the analysis. Lossy formats such as “.mp3” and “.ogg” work well with the human ear but may be missing crucial data for the study. Additionally, some encodings create artifacts that could be misleading to any automatic analyzer. Despite this, the ubiquity of the “.mp3” has meant much research in the field involves these as the source material. Increasingly, metadata mined from the web is incorporated in MIR for a more rounded understanding of the music within its cultural context, and this recently consists of the analysis of social tags for music.

1.5.7 **Feature representation**

Analysis can often require some summarising,[3] and for music (as with many other forms of data) this is achieved by feature extraction, especially when the audio content itself is analyzed and machine learning is to be applied. The purpose is to reduce the sheer quantity of data down to a manageable set of values so that learning can be performed within a reasonable time-frame. One common feature extracted is the Mel-Frequency Cepstral Coefficient (MFCC) which is a measure of the timbre of a piece of music. Other features may be employed to represent the key, chords, harmonies, melodies, main pitch, beats per minute or rhythm in the piece. There are a number of available audio feature extraction tools

1.5.8 **Statistics and machine learning**

Computational methods for classification, clustering, and modeling — musical feature extraction for mono and polyphonic music, similarity and pattern matching, retrieval

- Formal methods and databases — applications of automated music identification and recognition, such as score following, automatic accompaniment, routing and filtering for music and music queries, query languages, standards and other metadata or protocols for music information handling and retrieval, multi-agent systems, distributed search)
- Software for music information retrieval — Semantic Web and musical digital objects, intelligent agents, collaborative software, web-based search and semantic retrieval, query by humming / Search by

sound, acoustic fingerprinting
- Music analysis and knowledge representation — automatic summarization, citing, excerpting, downgrading, transformation, formal models of music, digital scores and representations, music indexing and metadata.

1.6 **Challenges**

It's estimated that 80% of the world's data is unstructured and not organized in a pre-defined manner. Most of this comes from text data, like emails, support tickets, chats, social media, surveys, articles, and documents. These texts are usually difficult, time-consuming and expensive to analyze, understand, and sort through.

Sentiment analysis systems allow companies to make sense of this sea of unstructured text by automating business processes, getting actionable insights, and saving hours of manual data processing, in other words, by making teams more efficient.

Some of the advantages of sentiment analysis include the

following: ● **Scalability**:

Can you imagine manually sorting through thousands of tweets, customer support conversations, or customer reviews? There's just too much data to process manually. Sentiment analysis allows processing data at scale in an efficient and cost-effective way.

● **Real-time analysis**:

We can use sentiment analysis to identify critical information that allows situational awareness during specific scenarios in real-time. Is there a PR crisis in social media about to burst? An angry customer is about to churn? A sentiment analysis system can help you immediately identify these kinds of situations and take action.

- **Consistent criteria**:

Humans don’t observe clear criteria for evaluating the sentiment of a piece of text. It’s estimated that different people only agree around 60-65% of the time when judging the sentiment for a particular piece of text. It’s a subjective task that is heavily influenced by personal experiences, thoughts, and beliefs. By using a centralized sentiment analysis system, companies can apply the same criteria to all of their data. This helps to reduce errors and improve data consistency.

1.7 **Motivations**

As mentioned in the Challenges, we decided to perform the sentiments of the singer in the music as the sentiment analysis of the music in this era is yet to be done. Our motivation to analyze the sentiment of the singer arrives as a song can be sung by different singers. According to the background score of the music and depending upon the musician the sentiment can be changed of the song. For example, a sad song can be manipulated by the musicians into a happy song. This is trivial in the field of Sentimental analysis of the song, which makes us accept the challenges and provide us the motivation to work on. we propose a novel approach for emotion classification of audio conversation based on both speech and text. The novelty in this approach is in the choice of features and the generation of a single feature vector for classification. Our main intention is to increase the accuracy of emotion classification of speech by considering both audio and text features.

1.8 **Hypothesis**

1.8.1 **Hypothesis 1**

The dataset that contains the audio samples has been annotated with its gender categorized as Male and Female and emotions categorized as neutral, calm, happy, sad, angry, fearful, disgust and surprised. The audio in training is classified on the basis of features in different clusters.

Unknown audio samples are provided results that provide the clusters of the emotion of the sample.

1.8.2 **Hypothesis 2**

The dataset for evaluation was done using the Rabindra Sangeet. The songs were crawled from different websites that has stored the Rabindra Sangeet. Each song was manually annotated about its emotion categorized as calm, happy, sad, angry and fearful. The audio in evaluation was classified according to the emotions.

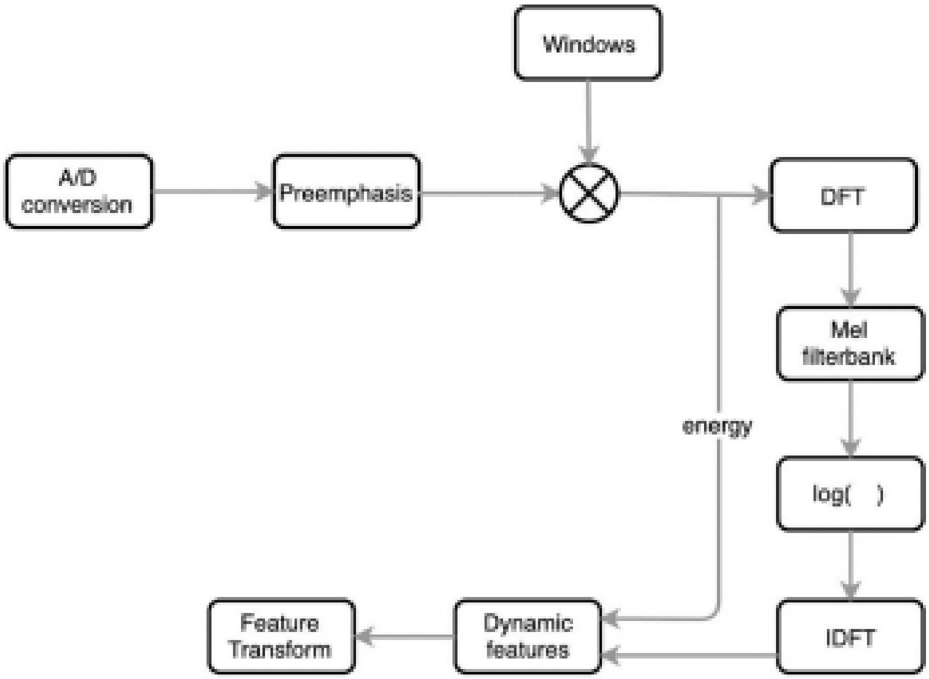
1.9 **Thesis Outline**

Our thesis consists of 7 chapters. Chapter 1 consists of the Introduction of our proposed work. Chapter 2 consists of the tools we have used for achieving the target of the work. Chapter 3 consists of Previous work done in the field. Chapter 4 consists of the preparation of our Dataset. Chapter 5 consists of the Implementation of our proposed work. Chapter 6 consists of the Evaluation part. Chapter 7 consists of Conclusion and Future Work.

METHODOLOGY

2.1MFCC

Below is the flow of extracting the MFCC features.

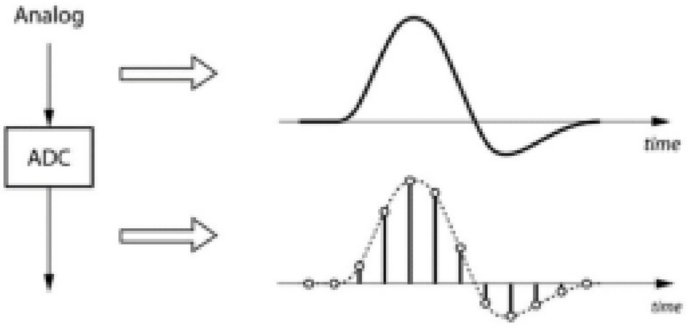


The key objectives are:

- Remove vocal fold excitation (F0) — the pitch information.
- Make the extracted features independent.
- Adjust to how humans perceive loudness and frequency of sound.
- Capture the dynamics of phones (the context).
- Mel-frequency cepstral coefficients (MFCC)
- Let's cover each step one at a time.

2.1.1 **A/D conversion**

A/D conversion samples the audio clips and digitizes the content, i.e. converting the analog signal into discrete space. A sampling frequency of 8 or 16 kHz is often used.

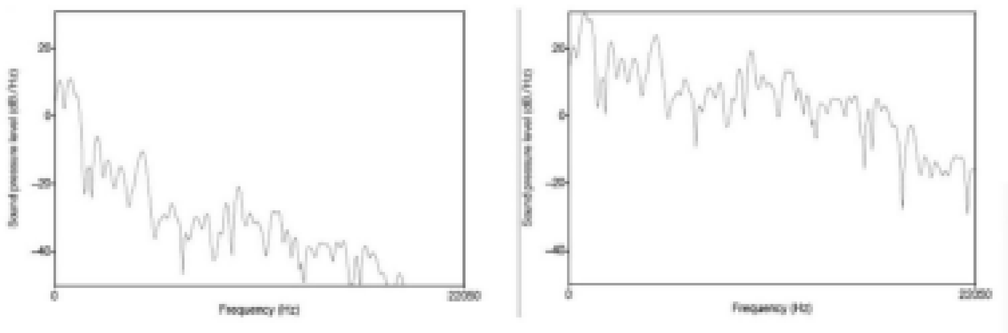


2.1.2 **Pre-emphasis**

Pre-emphasis boosts the amount of energy in the high frequencies. For voiced segments like vowels, there is more energy at the lower frequencies than the higher frequencies. This is called spectral tilt which is related to the glottal

source (how vocal folds produce sound). Boosting the high-frequency energy makes information in higher formants more available to the acoustic model. This improves phone detection accuracy. For humans, we start having hearing problems when we cannot hear these high-frequency sounds. Also, the noise has a high frequency. In the engineering field, we use pre-emphasis to make the system less susceptible to noise introduced in the process later. For some applications, we just need to undo the boosting at the end.
Pre-emphasis uses a filter to boost higher frequencies. Below is the before and after signal on how the high-frequency signal is boosted.

$$x'[t_d] = x[t_d] - \alpha x[t_d - 1] \qquad 0.95 < \alpha < 0.99$$



2.1.3 **Windowing**

Windowing involves the slicing of the audio waveform into sliding frames.

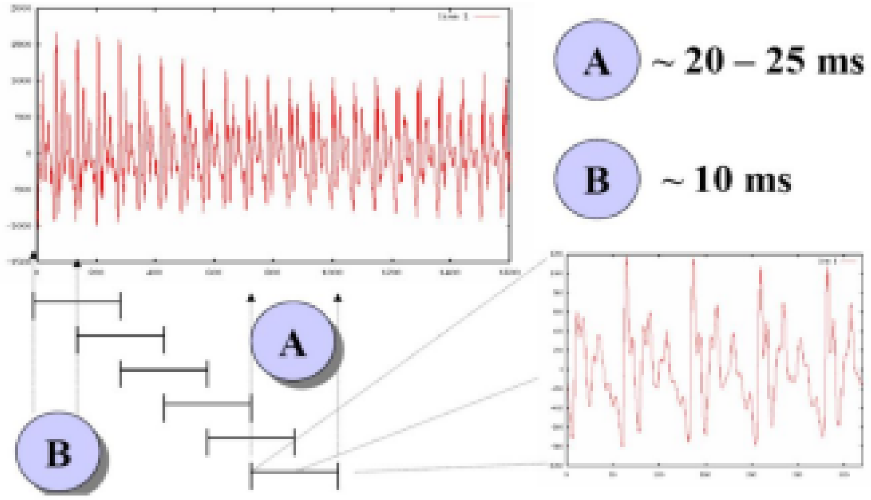
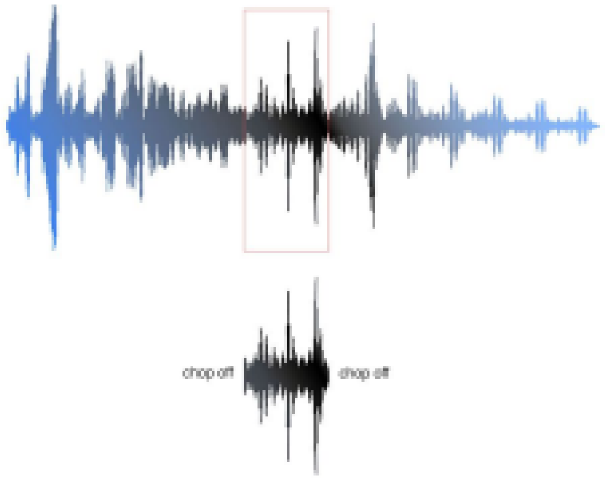


Image from Bryan Pellom

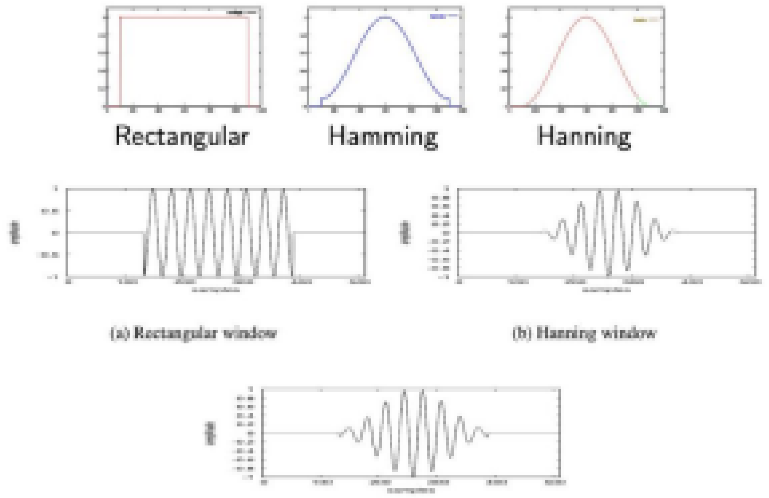
But we cannot just chop it off at the edge of the frame. The suddenly fallen in amplitude will create a lot of noise that shows up in the high-frequency. To slice the audio, the amplitude should gradually drop off near the edge of a frame.



Let’s say w is the window applied to the original audio clip in the time domain.

$$\underset{\text{sliced frame}}{x[n]} = w[n]\,\underset{\text{original audio clip}}{s[n]}$$

A few alternatives for w are the Hamming window and the Hanning window. The following diagram indicates how a sinusoidal waveform will be chopped off using these windows. As shown, for Hamming and Hanning window, the amplitude drops off near the edge. (The Hamming window has a slight sudden drop at the edge while the Hanning window does not.)



(a) Rectangular window

(b) Hanning window

(c) Hamming window

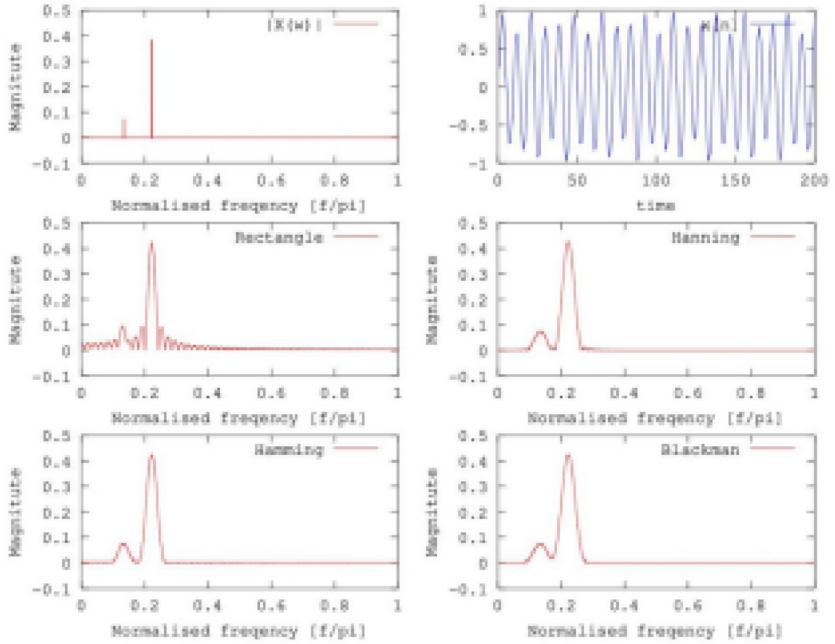
(Taylor, fig 12.1)

The corresponding equations for w are:

Hamming ($\alpha = 0.46164$) or *Hanning* ($\alpha = 0.5$) window

$$w[n] = (1-\alpha) - \alpha \cos\left(\frac{2\pi n}{L-1}\right) \qquad L : \text{ window width}$$

On the top right below is a soundwave in the time domain. It mainly composes of two frequencies only. As shown, the chopped frame with Hamming and Hanning maintains the original frequency information better with less noise compared to a rectangle window.



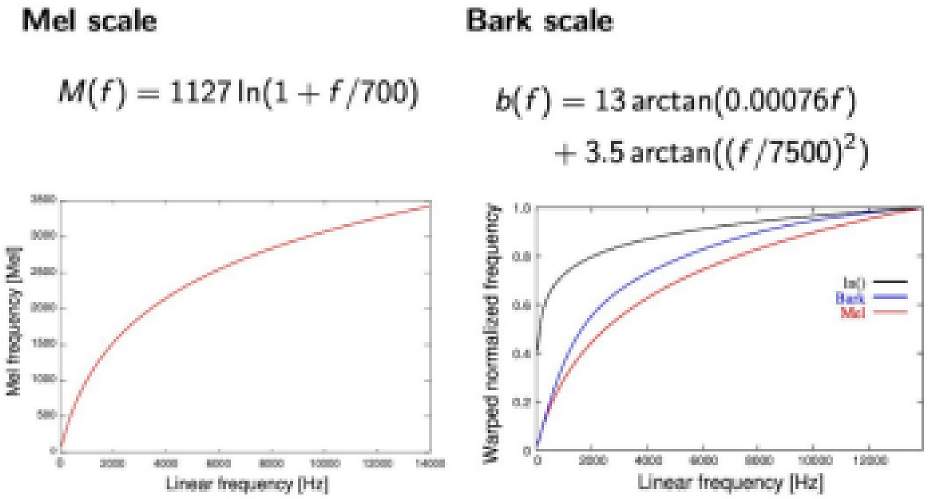
2.1.4 **Discrete Fourier Transform (DFT)**

Next, we apply DFT to extract information in the frequency domain.

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j\frac{2\pi}{N}kn\right)$$

2.1.5 **Mel filterbank**

As mentioned in the previous article, the equipment measurements are not the same as our hearing perception. For humans, the perceived loudness changes according to frequency. Also, perceived frequency resolution decreases as frequency increases. i.e. humans are less sensitive to higher frequencies. The diagram on the left indicates how the Mel scale maps the measured frequency to that we perceived in the context of frequency resolution.



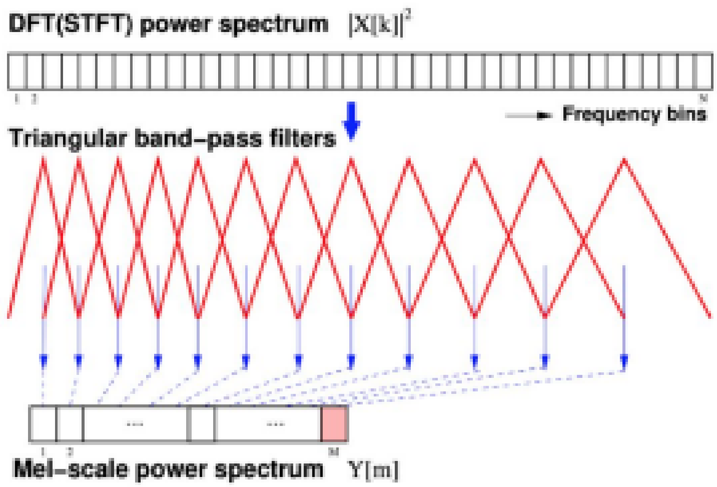
Mel scale

$$M(f) = 1127\ln(1 + f/700)$$

Bark scale

$$b(f) = 13\arctan(0.00076f) + 3.5\arctan((f/7500)^2)$$

All these mappings are non-linear. In feature extraction, we apply triangular band-pass filters to coverts the frequency information to mimic what a human perceived.


DFT(STFT) power spectrum $|X[k]|^2$
1 2
N
Frequency bins
Triangular band–pass filters
...
...
1 2
M
Mel–scale power spectrum $Y[m]$

First, we square the output of the DFT. This reflects the power of the speech at each frequency ($x[k]^2$) and we call it the DFT power spectrum. We apply these triangular Mel-scale filter banks to transform it into a Mel-scale power spectrum. The output for each Mel-scale power spectrum slot represents the energy from a number of frequency bands that it covers. This mapping is called the Mel Binning. The precise equations for slot m will be:

$$Y_t[m] = \sum_{k=1}^{N} W_m[k]\,|X_t[k]|^2$$

$$\text{where} \quad \begin{aligned} k &: \text{DFT bin number } (1, \ldots, N) \\ m &: \text{mel-filter bank number } (1, \ldots, M) \end{aligned}$$

The Trainangular bandpass is wider at the higher frequencies to reflect human hearing is less sensitivity in high frequency. Specifically, it is linearly spaced below 1000 Hz and turns logarithmically afterward.
All these efforts try to mimic how the basilar membrane in our ear senses the vibration of sounds. The basilar membrane has about 15,000 hairs inside the cochlear at birth. The diagram below demonstrates the frequency response of those hairs. So the curve-shape response below is simply approximated by triangles in Mel filterbank.

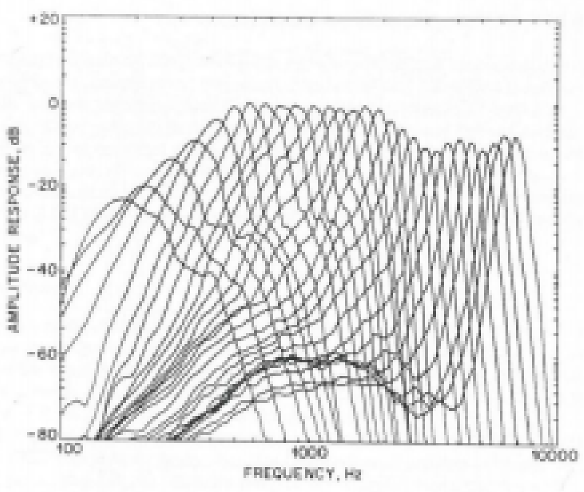


Figure 3.50 Frequency response curves of a cat's basilar membrane (after Ghitza [13]).

We imitate how our ears perceive sound through those hairs. In short, it is modeled by the triangular filters using Mel filtering bank.

2.1.5 **Log**

Mel filterbank outputs a power spectrum. Humans are less sensitive to small energy changes at high energy than small changes at a low energy level. In fact, it is logarithmic. So our next step will take the log out of the output of the Mel filterbank. This also reduces the acoustic variants that are not significant for speech recognition. Next, we need to address two more requirements. First, we need to remove the F0 information (the pitch) and make the extracted features independent of others.

2.1.6 **Cepstrum — IDFT**

Below is the model of how speech is produced.

Our articulations control the shape of the vocal tract. The source-filter model combines the vibrations produced by the vocal folds with the filter created by our articulations. The glottal source waveform will be suppressed or amplified at different frequencies by the shape of the vocal tract.

Cepstrum is the reverse of the first 4 letters in the word “spectrum”. Our next step is to compute the Cepstral which separates the glottal source and the filter. Diagram (a) is the spectrum with the y-axis being the magnitude. Diagram (b) takes the log of the magnitude. Look closer, the wave fluctuates about 8 times between 1000 and 2000. Actually, it fluctuates about 8 times for every 1000 units. That is about 125 Hz — the source vibration of the vocal folds.

As observed, the log spectrum (the first diagram below) composes of information related to the phone (the second diagram) and the pitch (the third diagram). The peaks in the second diagram identify the formants that distinguish phones. But how can we separate them?

Recall that periods in the time or frequency domain is inverted after transformation.

Recall that the pitch information has short periods in the frequency domain. We can apply the inverse Fourier Transformation to separate the pitch information from the formants. As shown below, the pitch information will show up on the middle and the right side. The peak in the middle is actually corresponding to F0 and the phone-related information will locate in the far left.

Here is another visualization. The solid line on the left diagram is the signal in the frequency domain. It is composed of the phone information drawn in the dotted line and the pitch information. After the IDFT (inverse Discrete Fourier Transform), the pitch information with the 1/T period is transformed to a peak near T at the right side.

So for speech recognition, we just need the coefficients on the far left and discard the others. In fact, MFCC just takes the first 12 cepstral values. There is another important property related to these 12 coefficients. The log power spectrum is real and symmetric. Its inverse DFT is equivalent to a discrete cosine transformation (DCT).

DCT is an orthogonal transformation. Mathematically, the transformation produces uncorrelated features. Therefore, MFCC features are highly unrelated. In ML, this makes our model easier to model and to train. If we model these parameters with multivariate Gaussian distribution, all the non-diagonal values in the covariance matrix will be zero. Mathematically, the output of this stage is

The following is the visualization of the 12 Cepstrum coefficients.

2.1.7 **Dynamic features (delta)**

MFCC has 39 features. We finalize 12 and what are the rest. The 13th parameter is the energy in each frame. It helps us to identify phones.

In pronunciation, context and dynamic information are important. Articulations, like stop closures and releases, can be recognized by the formant transitions. Characterizing feature changes over time provides the context information for a phone. Another 13 values compute the delta values d(t) below. It measures the changes in features from the previous frame to the next frame. This is the first-order derivative of the features.

The last 13 parameters are the dynamic changes of d(t) from the last frame to the next frame. It acts as the second-order derivative of c(t).

So the 39 MFCC features parameters are 12 Cepstrum coefficients plus the energy term. Then we have 2 more sets corresponding to the delta and the double delta values.

2.1.8 **Cepstral mean and variance normalization**

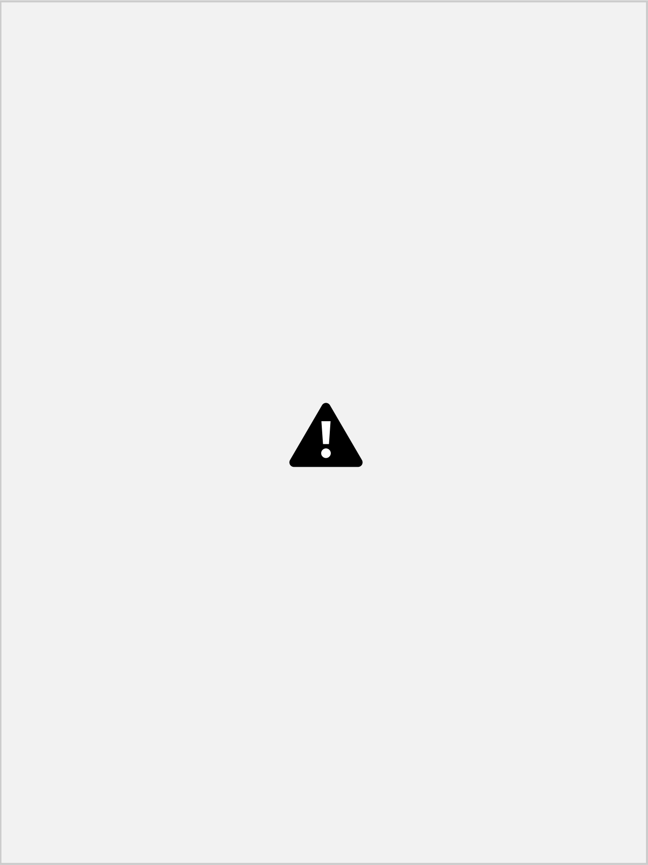
Next, we can perform the feature normalization. We normalize the features with its mean and divide it by its variance. The mean and variance are computed with the feature value j over all the frames in a single utterance. This allows us to adjust values to countermeasure the variants in each recording.

However, if the audio clip is short, this may not be reliable. Instead, we may compute the average and variance values based on speakers, or even over the entire training dataset. This type of feature normalization will effectively cancel the pre-emphasis done earlier. That is how we extract MFCC features. As a last note, MFCC is not very robust against noise.

2.2 **Neural Network**

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus a neural network is either a biological neural network, made up of real biological neurons or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by weight and summed. This activity is referred to as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be −1 and 1.

These artificial networks may be used for predictive modeling, adaptive control, and applications where they can be trained via a dataset. Self-learning resulting from experience can occur within networks, which can derive conclusions from a complex and seemingly unrelated set of information.



Neural networks are a set of algorithms, modeled loosely after the human brain, that is designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

Neural networks help us cluster and classify. We can think of them as a clustering and classification layer on top of the data you store and manage. They help to group unlabeled data according to similarities among the example inputs, and they classify data when they have a labeled dataset to train on. Neural networks can also extract features that are fed to other algorithms for clustering and classification; so you can think of deep neural networks as components of larger machine-learning applications involving algorithms for reinforcement learning, classification and, regression.

Deep learning maps inputs to outputs. It finds correlations. It is known as a "universal approximator", because it can learn to approximate an unknown function f(x) = y between any input x and any output y, assuming they are related at all (by correlation or causation, for example). In the process of learning, a neural network finds the right f, or the correct manner of transforming x into y, whether that be f(x) = 3x + 12 or f(x) = 9x - 0.1. Here are a few examples of what deep learning can do.

2.2.1 **Classification**

All classification tasks depend upon labeled datasets, that is, humans must transfer their knowledge to the dataset in order for a neural network to learn the correlation between labels and data. This is known as supervised learning.

Using the Classification algorithm we can Detect faces, identify people in images, recognize facial expressions (angry, joyful). Identify objects in images (stop signs, pedestrians, lane markers). Recognize gestures in video Detect voices, identify speakers, transcribe speech to text, recognize sentiment in voice Classify text as spam (in emails), or fraudulent (in insurance claims); recognize sentiment in text (customer feedback) Any labels that humans can generate, any outcomes that you care about and which correlate to data, can be used to train a neural network.

2.2.2 **Clustering**

Clustering or grouping is the detection of similarities. Deep learning does not require labels to detect similarities. Learning without labels is called unsupervised learning. Unlabeled data is the majority of data in the world. One law of machine learning is: the more data an algorithm can train on, the more accurate it will be. Therefore, unsupervised learning has the potential to produce highly accurate models.

With that brief overview of deep learning use cases, let's look at what neural nets are made of.

2.2.3 **Neural Network Elements**

Deep learning is the name we use for "stacked neural networks", that is, networks composed of several layers.

The layers are made of nodes. A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby

assigning significance to inputs with regard to the task the algorithm is trying to learn; e.g. which input is most helpful is classifying data without error? These input-weight products are summed and then the sum is passed through a node's so-called activation function, to determine whether and to what extent that signal should progress further through the network to affect the ultimate outcome, say, an act of classification. If the signals pass through, the neuron has been "activated."

Here's a diagram of what one node might look like.

A node layer is a row of those neuron-like switches that turn on or off as the input is fed through the net. Each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving your data.

Pairing the model’s adjustable weights with input features is how we assign significance to those features with regard to how the neural network classifies and clusters input.

2.2.4 Key Concepts of Deep Neural Networks

Deep-learning networks are distinguished from the more commonplace single-hidden-layer neural networks by their depth; that is, the number of node layers through which data must pass in a multistep process of pattern recognition.

Earlier versions of neural networks such as the first perceptrons were shallow, composed of one input and one output layer, and at most one hidden layer in between. More than three layers (including input and output) qualifies as “deep” learning. So deep is not just a buzzword to make algorithms seem like they read Sartre and listen to bands you haven’t heard of yet. It is a strictly defined term that means more than one hidden layer.

In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer’s output. The further you advance into the neural net, the more complex the features your nodes can recognize since they aggregate and recombine features from the previous layer.

Deep-learning networks perform automatic feature extraction without human intervention, unlike most traditional machine-learning algorithms. Given that feature, extraction is a task that can take teams of data scientists years to accomplish, deep learning is a way to circumvent the chokepoint of limited experts. It augments the powers of small data science teams, which by their nature do not scale.

When training on unlabeled data, each node layer in a deep network learns features automatically by repeatedly trying to reconstruct the input from which it draws its samples, attempting to minimize the difference between the network's guesses and the probability distribution of the input data itself. Restricted Boltzmann machines, for example, create so-called reconstructions in this manner.

In the process, these neural networks learn to recognize correlations between certain relevant features and optimal results – they draw connections between feature signals and what those features represent, whether it be a full reconstruction, or with labeled data.

A deep-learning network trained on labeled data can then be applied to unstructured data, giving it access to much more input than machine-learning nets. This is a recipe for higher performance: the more data a net can train on, the more accurate it is likely to be. (Bad algorithms trained on lots of data can outperform good algorithms trained on very little.) Deep learning's ability to process and learn from huge quantities of unlabeled data gives it a distinct advantage over previous algorithms.

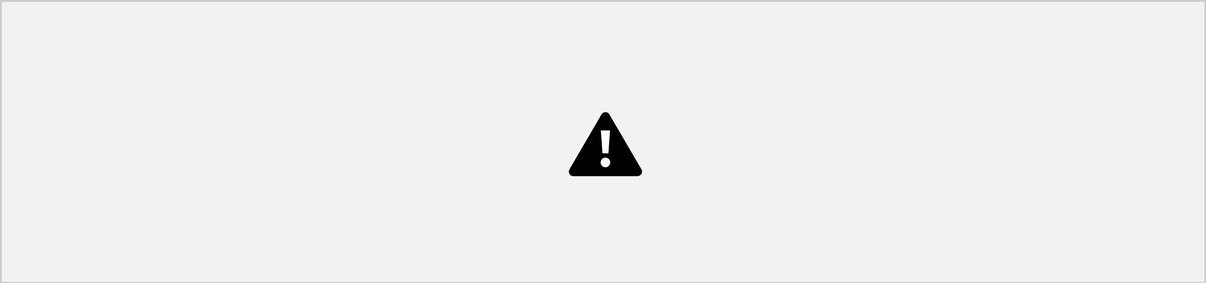
2.3 **Precision**

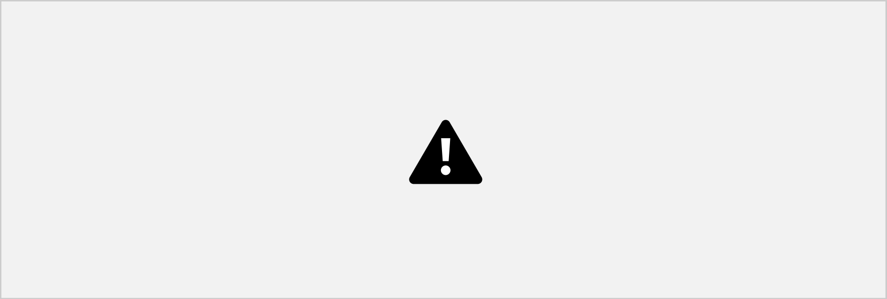
In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the query:

For example, for a text search on a set of documents, precision is the number of correct results divided by the number of all returned results.
Precision takes all retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. This measure is called *precision at n* or *P@n*.
Precision is used with recall, the percent of *all* relevant documents that are returned by the search. The two measures are sometimes used together in the F1 Score (or f-measure) to provide a single measurement for a system.

Precision is a good measure to determine when the costs of False Positive is high. For instance, email spam detection. In email spam detection, a false positive means that an email that is non-spam (actual negative) has been identified as spam (predicted spam). The email user might lose important emails if the precision is not high for the spam detection model.

This model predicts positive, to predict how often it is correct.





Precision helps when the costs of false positives are high. So let’s assume the problem involves the detection of skin cancer. If we have a model that has very low precision, then many patients will be told that they have melanoma, and that will include some misdiagnoses. Lots of extra tests and stress are at stake. When false positives are too high, those who monitor the results will learn to ignore them after being bombarded with false alarms.

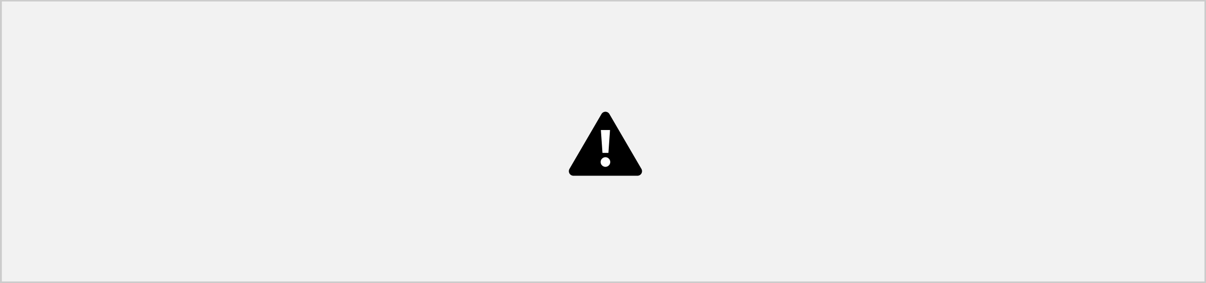
2.4**Recall**

In information retrieval, recall is the fraction of the relevant documents that are successfully retrieved.

For example, for a text search on a set of documents, recall is the number of correct results divided by the number of results that should have been returned.

In binary classification, recall is called sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query. It is trivial to achieve a recall of 100% by returning all documents in response to any query. Therefore, recall alone is not enough but one needs to measure the number of non-relevant documents also, for example by also computing the precision.

Recall helps when the cost of false negatives is high. What if we need to detect incoming nuclear missiles? A false negative has devastating consequences. Get it wrong and we all die. When false negatives are frequent, you get hit by the thing you want to avoid. A false negative is when you decide to ignore the sound of a twig breaking in a dark forest, and you get eaten by a bear. (A false positive is staying up all night sleepless in your tent in a cold sweat listening to every shuffle in the forest, only to realize the next morning that those sounds were made by a chipmunk. Not fun.) If you had a model that let in nuclear missiles by mistake, you would want to throw it out. If you had a model that kept you awake all night because *chipmunks*, you would want to throw it out, too. If like most people, you prefer to not get eaten by the bear, and also not stay up all night worried about chipmunk alarms, then you need to optimize for an evaluation metric that's a combined measure of precision and recall.



So Recall actually calculates how many of the Actual Positives our model capture through labeling it as Positive (True Positive). Applying the same understanding, we know that Recall shall be the model metric we use to select our best model when there is a high cost associated with False Negative.

Often, we think that precision and recall both indicate the accuracy of the model. While that is somewhat true, there is a deeper, distinct meaning of each of these terms. Precision means the percentage of your results that are relevant. On the other hand, recall refers to the percentage of total relevant results correctly classified by your algorithm. Undoubtedly, this is a hard concept to grasp in the first go.

2.5**F1 Score**

In the statistical analysis of binary classification, the **F_1 score** (also **F-score** or **F-measure**) is a measure of a test's accuracy. It considers both the precision *p* and the recall *r* of the test to compute the score: *p* is the number of correct positive results divided by the number of all positive results returned by the classifier, and *r* is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F_1 score is the harmonic mean of the precision and recall, where an F_1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

F1 is an overall measure of a model's accuracy that combines precision and recall, in that weird way that addition and multiplication just mix two ingredients to make a separate dish altogether. That is, a good F1 score means that you have low false positives and low false negatives, so you're correctly identifying real threats and you are not disturbed by false alarms. An F1 score is considered perfect when it's 1, while the model is a total failure when it's 0. The traditional F-measure or balanced F-score (**F_1 score**) is the harmonic mean of precision and recall:

The general formula for positive real β, where β is chosen such that recall is considered β times as important as precision, is:

The formula in terms of Type I and type II errors:

Two commonly used values for β are those corresponding to the F2 measure, which weighs recall higher than precision (by placing more emphasis on false negatives), and the F0.5 measure, which weighs recall lower than precision (by attenuating the influence of false negatives).

The F-measure was derived so that "measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision". It is based on Van Rijsbergen's effectiveness measure

Their relationship is where

The F_1 score is also known as the Sørensen–Dice coefficient or Dice similarity coefficient (DSC).

LITERATURE SURVEY

Music emotion recognition (MER) is a young, but the fast-expanding field, stimulated by the interest from the music industry to improve automatic music categorization methods for large-scale online music collections. An analysis of written music queries from creative professionals showed that 80% of the queries for production music contain emotional terms, making them one of the most salient and important components of exploratory music search. In the last decade, many new MER methods have been proposed.

However, methodological differences in data representation result in a choice of different evaluation metrics, which makes the accuracy of the algorithms impossible to compare. A wide variety of categorical and dimensional emotional models are used, such as basic emotions, valence and arousal model, Geneva Emotional Music Scales (GEMS), or custom mood clusters. Despite differences in data representation, most of the methods are essentially solving the same problem of mapping acoustic features (or lyrics and meta-data based features) to the emotional annotations. A specific learning algorithm can not always be adapted to other representations (though many algorithms, such as SVM or different types of neural networks, are versatile), but audio features are more often transferable. A benchmark can, therefore, enable a comparison of different methods and feature sets.

Humans are inherent with emotions or in other words, emotions are what make us human. The affective aspect of music also referred to as music mood has been recently recognized as an important aspect of organizing and obtaining music information.

It has been suggested in the paper that emotion models can be generally classified into two categories

1. Parametric Classification model: Emotions can be defined according to one or more dimensions. These parametric models of emotion endeavor to classify human emotions by locating where these lie in two dimensions (valence-arousal) or three dimensions (valence-arousal-intensity).

    a) Russell’s Circumplex Model of effect: This is a well known two-dimensional model called the Valence-Arousal Model proposed by Russell in 1980. This model assigns one axis to represent the Arousal level indicating the intensity in the form of high (active) and low values (inactive) and the other axis to represent Valence, which is an assessment of polarity ranging from positive (happy) to negative (sad).

    b) Thayer’s model: This model implements the theory that mood can be derived from two factors: Energy (High/ Low) and Stress (Positive/Negative), thus taxonomy is divided into four clusters: Anxious,

Contentment, Depression and Exuberance. The MIREX mood-taxonomy follows Thayer’s model of emotion is widely used for emotion analysis in various languages.

2. Categorical Classification model: This categorical model is based on Rusell’s model of the effect that describes four bipolar dimensions spaced 45◦ apart

    a. Positive Affect (excited/sluggish)
    b. Pleasantness (happy/sad)
    c. Negative Affect (distressed/relaxed)
    d. Engagement (aroused/still)
    e.

These quarters can be regarded as two pivots that decide the positive and negative affectivity of a person. Many research works for sentiment analysis of Twitter data have also adopted this model.

Using the taxonomies described, an extensive amount of work has been done on mood classification using audio, lyrics, social tags or the combination of either of these.

These features have been selected and accordingly various machine learning approaches were employed to classify the emotions.

As seen, I conducted a study on English songs based on hybrid features i.e. lyrics, audio and social tags. Audio and lyrics were used to build the classifiers while social tags for giving ground truth labels to the dataset. Using last.FM tagset, tags were collected and filtered to obtain 18 relevant tags i.e. 18 mood categories. Songs with only English lyrics were selected. The SVM classifier was used on lyric-only, audio-only and hybrid features (late fusion and feature concatenation) and compared. It was seen that among all systems, the hybrid system using late fusion accomplished the best performance outperformed audio-only system by 9.6%.

As seen worked upon the English-song dataset which was classified into emotion categories based on a weighted combination of extracted lyrical and audio features. Using Russell's Model of Affect, the features that classify songs were placed along two axes. Social tags from last.fm were collected and distributed amongst 9 mood categories. Feature weighting was performed as audio features were included too. The kNN classifier is improved as the Euclidean distance formula was modified to incorporate weights for each feature. This resulted in an accuracy of 83% in the test-set. One song can be classified into any number of classes but it must have a sufficient number of neighbors in those classes hence fuzzy classification was also applied. For a test set of 795 songs, the accuracy of 83.20% was observed by comparing the output to the classes derived from social tags.

Musical compositions utilize the five elements of music (rhythm, melody, pitch, harmony and interval), which play a significant role in human physiological and psychological functions, thus creating alterations in the mood.

Another problem of MER is that due to audio copyright restrictions, the data sets used in various studies are seldom made public and reused in other studies. Annotations are often obtained by crawling the tags from social music websites, such as last.fm or allmusic.com. In this case, the audio is usually copyrighted and can not be redistributed by the researchers. The music that is distributed for free under a license such as Creative Commons usually is less well-known and has fewer tags, and therefore needs to be annotated. Annotating with emotional labels is burdensome because with such a subjective task many annotations are needed for every item.

A fundamental property of music is that it unfolds over time. An emotion expressed in the song may also change over time, though it is always possible to reduce this variety to a single average value. The online music websites, such as moodfuse.com, musicovery.com, allmusic.com, usually represent songs in a mood space by a single label, which is always an approximation of the emotional content of the song. In the design of the benchmark, we recognized the time-dependent nature of music by setting out to predict the emotion of the music dynamically (per-second), i.e., the main purpose of the benchmark is to compare dynamic MER algorithms, also known as music emotion variation detection (MEVD) algorithms in the literature.

A fundamental property of music is that it unfolds over time. An emotion expressed in the song may also change over time, though it is always possible to reduce this variety to a single average value. The online music websites, such as moodfuse.com, musicovery.com, allmusic.com, usually represent songs in a mood space by a single label, which is always an approximation of the emotional content of the song. In the design of the benchmark, we recognized the time-dependent nature of music by setting out to predict the emotion of the music dynamically (per-second), i.e., the main purpose of the benchmark is to compare dynamic MER algorithms, also known as music emotion variation detection (MEVD) algorithms in the literature.

Since the late 1980s, time-varying responses to music were measured using Continuous Response Digital Interface. Usually, only one dimension (such as

tension, musical intensity or emotionality) was measured. Schubert proposed to use a two-dimensional interface (valence–arousal plane) to annotate music with emotion continuously. This approach was adopted by MER researchers as well.

The first study that models musical emotion unfolding over time with musical features (loudness, tempo, melodic contour, texture, and spectral centroid) was conducted by Schubert in 2004. The model, using linear regression, could explain from 33% to 73% of the variation in emotion. In 2006, Korhonen et al. suggested a method to model musical emotion as a function of musical features using system identification techniques. Korhonen et al. used the low-level spectral features extracted using Marsyas software (http://marsyas.info), and perceptual features extracted with PsySound software. The system reached a performance of 0.22 for valence and 0.78 for arousal in terms of the coefficient of determination (R2). In 2010, Schmidt et al. Author used Kalman filtering to predict per-second changes in the distribution of emotion overtime on 15-second music excerpts. In 2011, Schmidt and Kim suggested using a new method—Conditional Random Fields—to model continuous emotion with a resolution of 11 × 11 in valence–arousal space. A very small feature-set was used—MFCCs, spectral contrast and timbre—and the system reached the performance of 0.173 in terms of Earth Mover's Distance (between the true 11 × 11 2D histograms of valence–arousal values and predicted one). Panda et al. [24] used Support Vector Machines and features extracted with Marsyas and MIRToolbox to track music over quadrants of valence–arousal space. Imbrasaite et al. combined Continuous Conditional Random Fields with a relative representation of features. Later, Imbrasaite et al. showed that using Continuous Conditional Neural Fields offers an improvement over the previous approach. Wang et al. represented the ambiguity of emotion through a Gaussian distribution and tracked the emotion variation over time using a mapping between music emotion space and low-level acoustic feature space through a set of latent feature classes. Markov et al. used Gaussian Processes for dynamic MER. The bidirectional Long Short-Term Memory Recurrent Neural Networks was first applied to continuous emotion recognition not in the domain of music but in the domain of multimodal human emotion detection from speech, facial expression and shoulder gesture.

Most of the algorithms mentioned in this section were employed in the benchmark: Support Vector Regression, linear regression, Kalman filtering, Gaussian Processes, Conditional Random Fields, Continuous Conditional Neural Fields and Long Short-Term Memory Recurrent Neural Networks, giving us an opportunity to qualitatively compare their performance in the benchmark.

Most of the studies reviewed above did not release public data. The only exception is the MoodSwings dataset, developed by Schmidt et al., which comprises 240 segments of US pop songs (each 15-second long) with per-second VA annotations, collected through MTurk. After an automatic verification step that removed unreliable annotations, each clip in this dataset was annotated by 7 to 23 subjects.

A similar task from a different domain is continuous emotion recognition from human behavior. Audiovisual emotion challenge (AVEC) is a challenge that has been running since 2011 and is addressing the problem of continuous emotion recognition. Since 2011, they used SEMAINE and RECOLA databases which include human behavior with continuous emotion labels. There are also public datasets with static per song music emotion annotations. The DEAP dataset has the ratings on valence, arousal and dominance for 120 clips of one-minute music video clips of Western pop music. Each clip was annotated by 14–16 listeners (50% female), who were asked to rate the felt valence, arousal and dominance on a 9-point scale for each clip. The AMG1608 dataset contains the VA ratings for 1,608 Western music in different genres, also annotated through MTurk.

DATASET PREPARATION

In order to prepare the corpus, we have used the RAVDESS dataset (The Ryerson Audio-Visual Database of Emotional Speech and Song).

4.1 **Dataset Description**

The dataset consists of 1500 audio file input from 24 different speakers of which 12 male and 12 female speakers. They recorded the audio in 8 different emotions categorized as neutral, calm, happy, sad, angry, fearful, disgust and surprised.

4.2 **Dataset Annotation**

Dataset downloaded was annotated for the categorization for different groups, respective of the author. Dataset was divided into multiple numpy files respective of emotions and independently was divided into another two classes of Male and Female.

IMPLEMENTATION

We propose the Music Information system named RIEA for Information, Emotion Retrieval of Music, where the input is a song and output is the emotion of song and information of gender.

5.1**Dataset Preparation for training**

For the training of neural networks to learn the information and emotion we used the RAVDESS dataset (The Ryerson Audio-Visual Database of Emotional Speech and Song). The dataset consists of 1500 audio file input from 24 different speakers of which 12 male and 12 female speakers. They recorded the audio in 8 different emotions categorized as neutral, calm, happy, sad, angry, fearful, disgust and surprised.

5.2 **Feature Extraction of Audio**

1500 audio files were used to extract there different features of emotions. Using the Librosa toolkit we load each audio file using the hyperparameter of 3 seconds audio at a time, the sampling rate of 44100 Hz and offset of 0.5 and resample type of “Kaiser fast”. The process gives the output of the audio time series and the sampling rate of each audio time series. The array of audio time series was used for the extraction of the feature using MFCC. Hyperparameters used to retrieve the features were sample rate extracted from the output of the Librosa toolkit and the number of MFCC features was taken 13. The mean of MFCC features of each audio was taken for the classification process.

These process results give the dataset for training with labeled extracted features.

5.3**Training of the Model**

The labeled extracted feature was used as the input for the training of neural networks. Before providing input to the neural network 20% was split for validating the model and 80% was used for training of the model.

We used the Sequential model for training consists of 4 layers of 1 Dimensional Convolutional layers having hyperparameters of filters 512, 256, 256, 128 respectively, kernel size of 5, padding of "same" and Activation function of "relu". 1 Dimension Maxpooling with pool size 8 and Dropout with 0.5 was used. The final activation function of "softmax" was used.

"RMSprop" optimizer was used with a learning rate of 0.0001 and decay of "1e-6". The model loss was calculated using "Categorical Crossentropy".

For training batch size of 512 was used for 1000 epochs.

EVALUATION

In this chapter we discussed the experiment result, and evaluation.

6.1 **Experimental Results**

The system was tested with 184 unknown audio songs of different languages. The results are given below.

Orginal	**Predicted**	**Remarks**
female_fear fu l	female_fear fu l	True
male_sad	male_sad	True
female_calm	female_sad	False
male_sad	male_sad	True
female_happy	female_fear fu l	False
female_sad	female_angry	False
male_fearful	male_happy	False
female_fear fu l	female_sad	False
male_happy	male_happy	True
male_sad	male_sad	True
female_fear fu l	female_sad	False
female_happy	female_angry	False
male_fearful	male_sad	False

female_fearfu l	female_angry	False
male_fearful	male_happy	False
female_calm	female_calm	True
male_angry	male_angry	True
female_calm	female_calm	True
female_calm	female_calm	True
male_calm	male_sad	False
male_calm	male_calm	True
male_happy	male_calm	False
male_angry	male_angry	True
male_happy	male_happy	True
male_calm	male_sad	False
male_fearful	male_angry	False
male_angry	male_angry	True
female_happy	female_angry	False
male_fearful	male_fearful	True
female_fearfu l	female_angry	False
male_calm	male_sad	False
male_calm	male_calm	True
female_sad	female_calm	False
female_angry	female_angry	True
male_happy	male_happy	True
female_happy	female_happy	True
male_sad	male_happy	False

female_angry	female_angry	True
male_calm	male_calm	True

male_calm	male_sad	False
female_happy	female_happy	True
female_calm	female_calm	True
female_calm	female_fear fu l	False
male_fearful	male_fearful	True
female_angry	female_angry	True
male_sad	male_sad	True
male_fearful	female_fear fu l	False
male_fearful	male_angry	False
male_calm	male_calm	True
female_sad	female_calm	False
male_fearful	male_angry	False
male_sad	male_calm	False
female_fear fu l	female_angry	False
female_calm	female_sad	False
female_calm	female_calm	True
female_fear fu l	female_angry	False
male_sad	male_sad	True
male_fearful	male_fearful	True

male_happy	male_sad	False
female_sad	female_fear fu l	False
male_sad	male_sad	True
male_calm	male_calm	True
female_happy	female_happy	True
male_angry	male_angry	True
female_happy	female_fear fu l	False
female_sad	female_calm	False
male_angry	male_angry	True
female_fear fu l	male_sad	False
male_happy	male_happy	True
male_calm	male_sad	False
male_calm	male_calm	True
female_calm	female_calm	True
female_happy	female_fear fu l	False
male_calm	male_sad	False
female_sad	female_sad	True
male_happy	male_happy	True
female_fear fu l	female_angry	False
male_angry	male_angry	True
male_happy	male_happy	True

male_happy	male_angry	False
female_happy	female_fear fu l	False
female_angry	male_angry	False
male_calm	male_sad	False
female_sad	female_calm	False
female_happy	female_fear fu l	False
male_fearful	male_fearful	True
male_angry	male_fearful	False
male_happy	male_angry	False
male_happy	male_angry	False
male_angry	male_angry	True
male_fearful	male_fearful	True
female_sad	female_happy	False
male_angry	male_angry	True
female_happy	female_angry	False
male_sad	male_calm	False
male_fearful	male_sad	False
female_fear fu l	female_angry	False
male_angry	male_fearful	False
female_angry	female_angry	True
male_happy	male_angry	False
male_calm	male_sad	False

male_fearful	male_fearful	True
female_happy	female_angry	False
male_happy	male_fearful	False
male_fearful	male_fearful	True
male_happy	male_angry	False
male_calm	male_calm	True
female_happy	female_sad	False
female_fearfu l	female_angry	False
female_calm	female_sad	False
male_calm	male_sad	False
female_calm	female_calm	True
male_angry	male_angry	True
male_sad	male_sad	True
male_sad	male_sad	True
male_sad	male_sad	True
male_angry	male_angry	True
male_happy	male_angry	False
male_sad	male_angry	False
female_angry	female_angry	True
female_angry	male_angry	False
male_angry	male_angry	True
male_calm	male_sad	False
male_angry	male_angry	True
female_sad	female_calm	False
male_sad	male_sad	True

male_happy	male_happy	True
male_happy	male_happy	True
male_calm	male_sad	False
female_sad	female_sad	True
female_fear fu l	female_fear fu l	True
female_fear fu l	female_fear fu l	True
female_happy	female_happy	True
female_happy	female_calm	False
female_fear fu l	male_angry	False
female_sad	female_fear fu l	False
male_fearful	male_angry	False
male_fearful	male_angry	False
female_sad	female_sad	True
male_calm	male_sad	False
male_fearful	male_fearful	True
male_fearful	male_angry	False
female_sad	female_fear fu l	False
female_calm	female_calm	True
male_calm	male_calm	True
male_angry	male_angry	True
female_sad	female_calm	False

female_sad	female_sad	True
female_angry	female_angry	True
female_happy	female_fear fu l	False
male_calm	male_sad	False
male_happy	male_angry	False
female_fear fu l	female_fear fu l	True
female_calm	female_calm	True
male_happy	male_angry	False
female_happy	female_happy	True
female_happy	female_fear fu l	False
female_sad	female_sad	True
male_angry	male_angry	True
female_happy	female_angry	False
female_happy	female_happy	True
female_angry	female_angry	True
female_sad	female_fear fu l	False
male_calm	male_happy	False
female_sad	female_sad	True
female_sad	female_sad	True
male_happy	male_calm	False
male_calm	male_fearful	False

male_fearful	male_fearful	True
female_calm	female_sad	False
female_fearfu l	female_calm	False
male_angry	male_sad	False
female_sad	female_fearfu l	False
male_angry	male_angry	True
male_angry	male_angry	True
male_happy	male_happy	True
male_angry	male_fearful	False
female_angry	male_fearful	False
male_angry	male_angry	True
female_fearfu l	female_fearfu l	True
female_sad	female_sad	True
female_calm	female_calm	True
male_fearful	male_happy	False
female_angry	female_angry	True

6.2 **Evaluation Results**

The output of the system was evaluated using the precision, recall and f1 score of each category.

Precision Recall F1 Score

female_angry 0.38 0.73 0.50 **female_calm** 0.56 0.67 0.61
female_fearful 0.26 0.29 0.28 **female_happy** 0.86 0.30 0.44
female_sad 0.53 0.38 0.44

male_angry 0.49 0.81 0.61 **male_calm** 0.67 0.35 0.46
male_fearful 0.60 0.43 0.50 **male_happy** 0.64 0.43 0.51
male_sad 0.36 0.71 0.48

Confusion Matrix for the evaluation is shown below.

	female_ angry	**female _calm**	**female_f earful**	**female_ happy**	**female _sad**	**male_ angry**	**male_ calm**	**male_f earful**	**male_ happy**	**male _sad**
female_ angry	8	0	0	0	0	2	0	1	0	0
female_ calm	0	10	1	0	4	0	0	0	0	0
female_f earful	7	1	5	0	2	1	0	0	0	1
female_ happy	5	1	7	6	1	0	0	0	0	0
female_ sad	1	6	5	1	8	0	0	0	0	0
male_an gry	0	0	0	0	0	17	0	3	0	1
male_cal m	0	0	0	0	0	0	8	1	1	13
male_fe arful	0	0	1	0	0	6	0	9	3	2

male_happy	0	0	0	0	0	8	2	1	9	1
male_sad	0	0	0	0	0	1	2	0	1	10

We have evaluated our model with unknown data of Rabindra Sangeet and found the following result. Total number of samples was 92.

	angry	**calm**	**fearful**	**happy**	**sad**
angry	12	0	0	0	0
calm	29	4	2	0	0
fearful	7	1	8	0	2
happy	4	0	0	1	0
sad	14	0	1	0	7

Precision, Recall and f1 score of Rabindra Sangeet is given below.

precision recall f1-score

angry 0.18 1 0.31

calm 0.8 0.11 0.2

fearful 0.73 0.44 0.55

happy 1 0.2 0.33

sad 0.78 0.32 0.45

During the evaluation using Rabindra Sangeet we found the result of accuracy of 35%. The result found was expected as we ignored few of the parameters which have been discussed in Conclusion Section.

CONCLUSION & FUTURE WORK

We proposed an emotion detection system of songs. The system is based on an audio signature in WAV format. The performance of the system in below-average i.e 55% accuracy. The system can be further tuned to get a better result.
We evaluated the model on Rabindra Sangeet and found the accuracy of 35%. The main issue for the model is the system fails to produce an accurate result if the background music is taken into account.

In our future work, we proposed to create a system where the background music of the audio will be separated from the song and independently the two systems will be trained on the based of music and lyrics, respectively. This will produce a different scorer of the emotions which will combine together to get the emotion of the music and another layer of attention can be used which stores the emotion of textual lyrics.

BIBLIOGRAPHY

- S. Chauhan and P. Chauhan, "Music mood classification based on lyrical analysis of Hindi songs using Latent Dirichlet Allocation," *2016 International Conference on Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds*, Noida, 2016, pp. 72-76.
  doi: 10.1109/INCITE.2016.7857593

- S. Shukla, P. Khanna and K. K. Agrawal, "Review on sentiment analysis on music," *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, Dubai, 2017, pp. 777-780.
  doi: 10.1109/ICTUS.2017.8286111

- Aljanaki, Anna AND Yang, Yi-Hsuan AND Soleymani, Mohammad, "Developing a benchmark for emotional analysis of music,"
  doi: 10.1371/journal.pone.0173392

- Grekow, Jacek, "Audio features dedicated to the detection and tracking of arousal and valence in musical compositions," Journal of Information and Telecommunication, pp. 322-333
  doi: 10.1080/24751839.2018.1463749

- Youngmoo E. Kim and Erik M. Schmidt and Raymond Migneco and Brandon G. Morton and Patrick Richardson and Jeffrey J. Scott and Jacquelin A. Speck and Douglas Turnbull. “State of the Art Report: Music Emotion Recognition: {A} State of the Art Review, ” Proceedings of the 11th International Society for Music Information Retrieval Conference, {ISMIR} 2010, Utrecht, Netherlands, August 9-13, 2010, pp. - 255-266

- Wikipedia contributors. (2019, August 27). Multimodal sentiment analysis. In *Wikipedia, The Free Encyclopedia*. Retrieved 08:25, November 29, 2019, from https://en.wikipedia.org/w/index.php?title=Multimodal_sentiment_analysis&oldid=912780237

- http://nlpprogress.com/english/multimodal.html

- Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391. https://doi.org/10.1371/journal.pone.0196391.

- S, Maghilnan & M, Rajesh. (2018). Sentiment Analysis on Speaker Specific Speech Data.

- https://monkeylearn.com/sentiment-analysis/

- Jasmine Bhaskar, K. Sruthi, Prema Nedungadi, “Hybrid Approach for Emotion Classification of Audio Conversation Based on Text and Speech Mining,,” Procedia Computer Science, Volume 46, 2015, Pages 635-643, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.02.112.