

Customer Satisfaction Based on Voice Signals

Siyamankela Bomela

Thesis presented in fulfilment
of the requirements for the degree of
Bachelor of Science Honours
at the University of the Western Cape

Supervisor: Reg Dodds
Co-supervisor: Mehrdad Ghaziasgar

This version September 22, 2017

Declaration

I, SIYAMAMKELA BOMELA, declare that this thesis “*Customer Satisfaction Based on Voice Signals*” is my own work, that it has not been submitted before for any degree or assessment at any other university, and that all the sources I have used or quoted have been indicated and acknowledged by means of complete references.

Signature:

Date:

SIYAMAMKELA BOMELA.

Abstract

This paper follows the development of a system that recognises a users emotion based on their voice. The user will be recorded saying a phrase and based on that phrase the system will determine whether the user is happy, neutral or sad. This system will be developed by using machine learning techniques and audio feature extraction algorithms.

Key words

Mel frequency co-efficient cepstrum

Fast fourier transform

Support vector machine

Internet

Graphical user interface

Cross Validation

Model Selection

Dataset

Acknowledgment

This thesis is a compilation of the efforts of many people that helped and me through the years. I would first like to thank my supervisor Mr Dodds and my co-supervisor Mr Mehrdad Ghazi-asgar for encouraging me during my study. Without our weekly meetings, this work would not have been possible. At this time I would like to extend a very special thanks to the post-graduate assistant . Without their help I would certainly not be where I am today.

Contents

Declaration	iii
Abstract	v
Key words	vii
Acknowledgment	ix
Glossary	xiii
1. User Requirements Document	1
1.1 Introduction	1
1.2 Users' view of the problem	1
1.3 Description of the problem	2
1.4 Softwares' Limitations	3
2. Requirements Analysis Document	4
2.1 Introduction	4
2.2 Designers intepretation of the users requirements	4
2.3 Related Work	5
2.3.1 First Solution	5
2.3.2 Second Solution	7
2.3.3 Third Solution	7
2.3.4 Best Solution	7
2.4 Testing	8
3. System Design	9
3.1 Description of System Interface	9
3.1.1 System Functionality	10
3.2 High Level Design	11
3.3 Low Level Design	12
3.4 Feature Extraction	13
3.4.1 Mel Frequency Cepstrum Coefficient	13
3.4.1.1 Fast Fourier Transform(FFT)	15
3.4.1.2 Mel Scale Filtering	15
3.4.1.3 Logarithm	15
3.4.1.4 Discrete Cosine Function	15
3.5 Feature Classification	16
3.5.1 Support Vector Machines	16

4.	Code Documentation	17
4.1	Mel Frequency Cepstrum Co-efficients	17
4.1.1	Fast Fourier Transform	17
4.1.2	MelScale Filtering	17
4.1.3	Logarithm	18
4.1.4	Discrete Cosine Transform	18
4.1.5	Vectorization	19
4.2	Training	20
4.2.1	Support Vector Machine	20
4.3	Libraries used	20

Glossary

MFCC Mel frequency co-efficients cepstrum

FFT Fast fourier transform

GUI Graphical user interface

SVM Support Vector Machines

DCT Discrete Cosine Function

Chapter 1

User Requirements Document

1.1 Introduction

The following are ideas of how the user wants the system to function, what its expected outcomes are. After the users view a short and long description of the problem will be given and the expected limitation of the software.

1.2 Users' view of the problem

Emotion recognition is the ability to tell how a person is feeling based on the response on their face or from the tone of their voice, The aim of the project is to determine the satisfaction of a customer, whether the customer was happy or not with the service that he was offered. The customer satisfaction can be classified into three different categories; positive(which could be happy, excited, surprised), neutral(No emotion detected) or negative(sad, angry, disappointed).

The user requires a system that will be viable for all customers, whether a sale is done in the store or telephonically. The system should be able to determine what the overall emotions the customer went through during the sale.

The will also be a matter of accuracy of the results, as people have different signals of expressing their emotions. The system will have to be able to accommodate any user.

1.3 Description of the problem

The user require a system that will be able to determine whether s customer is happy with the service they have obtained. Numerous systems have been implemented for such a service, but these systems are mostly based on facial expressions. They use a camera to look at the changes in the customers face to determine emotion. The limitations of such systems are that the person has to be in the same room as the camera to determine the satisfaction.

Organizations such as call centers rely on voice to voice interaction to provide services or make sales, which means the system of facial expression cannot be used. The main problem that this paper aims to to tackle is to create a system that will be able to work for both kind of scenarios mentioned above, the system should accommodate sales made in store and sales that are made through call center sales agents.

The proposed system should be able to determine how satisfied a customer by a service. This will be done by observing the changes that the customer experiences as they interact with the sales person and how their emotions improve as they buy the product .

1.4 Softwares' Limitations

The software can only be implemented on one customer per time.

The system will only be used in an environment with minimum people and should be in environment where there is no noise and no busy people moving around constantly.

Chapter 2

Requirements Analysis Document

2.1 Introduction

The Requirements analysis document focuses on the designers point of view. This document will look at how the system is expected to function, what tools will be required for the implementation of the system and what factors should be considered in order to ensure the system functions fully.

The designer will also be expected to analyze the user requirements document in order to determine if any data is required, data that will assist in improving the functionality of the system or data that is required in order for the system to be usable.

2.2 Designers interpretation of the users requirements

Given the above user requirements, the system created will have to determine emotions through the customers voice, When a customer makes a sale the voice will be analyzed to determine their emotion. This system will be able to accommodate the requirements the user had.

The user will have to be in an environment with minimum noise in order for the system to clearly hear their voice, A more

detailed approach of the project can be seen as the steps given below:

- A high quality microphone: This will be used to get high quality audio samples from the user to be used when determining their emotions
- A pre-existing database of emotion recognition audio: This will be used for comparing the customers sample audio to that similar in the database
- Python programming language: The system will be built using this language

These are the basics needed in order to implement the project. The system will be designed to work on computers.

2.3 Related Work

2.3.1 First Solution

A paper written by Te Won Lee, Jiucang Hao, Kwokleung Chan and Oh-Wook Kwon from the university of Carlifonia in San Diego, which also focused on Emotion Recognition by Speech Signals was implemented successfully with a high accuracy.

While designing the system, they noticed that the most important aspect that improved the results was the pitch and energy that the user had while using the microphone. They used a hybrid o Quadratic Discriminant analysis algorithm, Support vector machines and MFCC as the base feature. Their chosen databases was the SUSAS database and the AIBO database. The SUSAS

database is text-independent, while the AIBO database is speaker independent.

The SUSAS database when tested came back with an accuracy of 96.3 percent for stressed and neutral emotions, whilst the AIBO database provided a 42.3 percent accuracy. (Kwon et al., 2003)

2.3.2 Second Solution

The Indian Institute of Technology Kharagpur wrote a paper on Emotion Recognition through speech. They used two different databases, one was acted emotion speech and the second one was real emotion speech samples. The project used the Gaussian Mixture Model. The average classification accuracy obtained was 95 percent. (Rao et al., 2003)

2.3.3 Third Solution

The Hewlett-Packard laboratories wrote a paper on a system called Recognition of Emotions in Interactive voice system. The system was designed to determine when a customer gets annoyed or angry by the Interactive Voice Response system, it would transfer the customer to a real human call center agent.

HP used two databases, one was obtained from actors which imagined a scenario and acted it out. The second database used was obtained from the linguistic department in the University of Pennsylvania.

They used Support Vector Machines and achieved an overall 77 percent accuracy. (Yacoub et al., 1990)

2.3.4 Best Solution

All the solutions provided excellent results, most of the excellence of the results mostly depended on which database they use and the method they chose to use. From all these solutions it is quite clear that in order to get great accuracy, one should use a hybrid for the classifiers and obtain a reliable database.

The best solution is the first solution. Their system was implemented the following way (Kwon et al., 2003):

1. Feature Extraction: They selected the log energies, pitch and MFCC as their base features, then selected their frame shift to be 10 milliseconds.
2. Feature Selection: Identifies features and properties of speech that are important for distinguishing different emotions.
3. Classification: In this stage they used their chosen algorithms to process the audio to obtain the emotion outcome

The reason for choosing this as the best solution is because it is more in line with the system the user requires and this system was tested for different emotions.

2.4 Testing

To determine whether the system is functioning properly, the following can be looked at

- Test if the system recognizes audio when the customer speaks
- Test if it can differentiate between angry, sad or neutral
- Test if it gives an output.

These are the basics tests that the system should pass to be seen as functional.

Chapter 3

System Design

3.1 Description of System Interface

The chosen interface for the system is a GUI. The GUI will be simply built to allow for easy input and for everyone be able to interact with the system. After the classification of the audio has been completed, the GUI will then provide the user with the emotion detected and the probability that the emotion detected is the correct one.

The system is composed of the following buttons and options:

- Directory Input line: Input the directory with the file that you would like to classify.
- Get Audio Button: This gets the .wav file that is in the directory input line, then processes it and tells you the emotion.
- Record Button: This records audio through the microphone and then tells you the emotion detected after recording.
- Progress label: Informs you of what the system is currently doing. e.g Recording, Processing, Complete
- Emotion Label: Informs of the detected label

- LCD Number Display: Shows the percentage of the accuracy of the predicted emotion.
- Exit Button: This allows the user to exits.

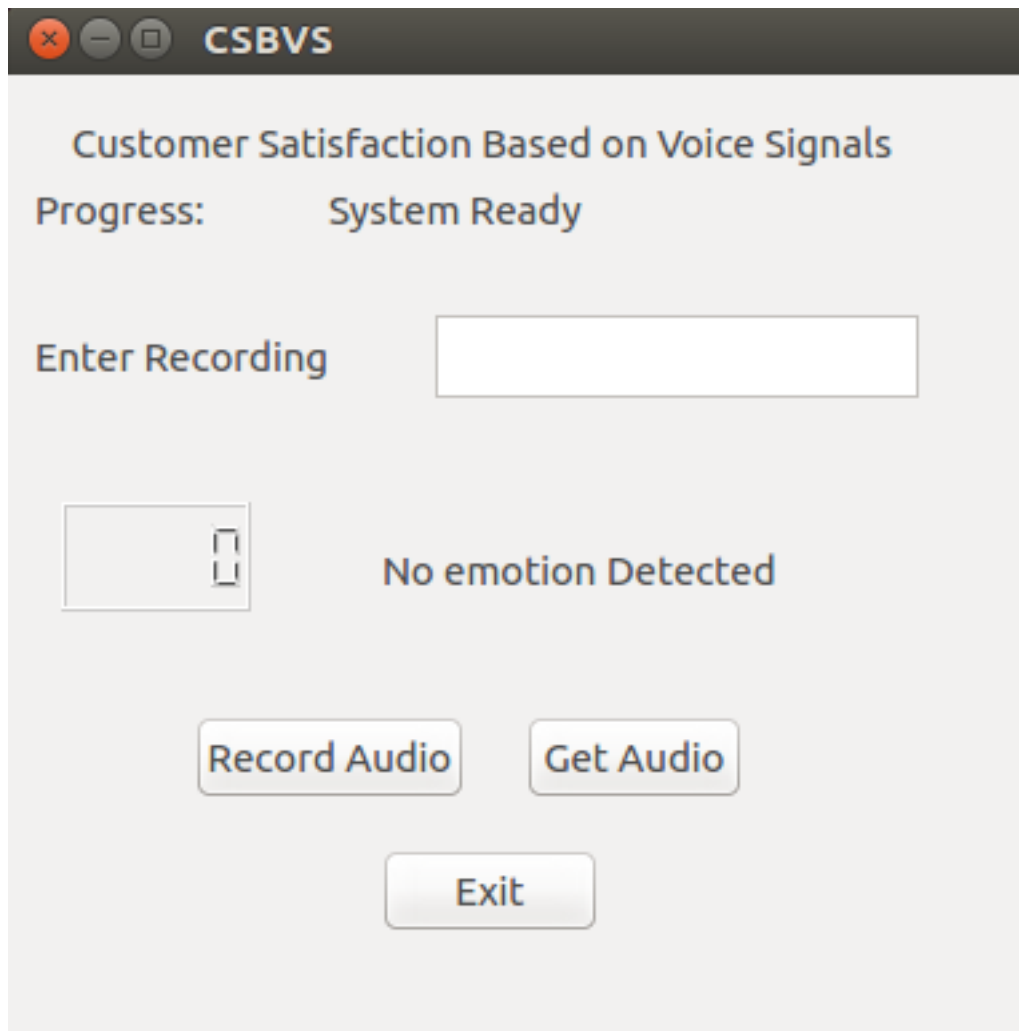


Figure 1: CSBVS User Interface

3.1.1 System Functionality

The system has a microphone that takes a short recording of the user, the audio file is saved in .wav format. If a pre-existing audio file had already been pre-recorded it can also be uploaded on the system. The system takes recordings of four seconds. The audio file also has to be of a mono channel.

The microphone has to be directly in front of a user when recording for best quality in sound to ensure a high accuracy when the audio is being classified.

The system will be able to distinguish between three emotion, which are anger/disappointment, happiness and neutral.

3.2 High Level Design

This section deals with how the architecture of the system is structured. A definition of how the each component will work and how the architecture will be structured.

The system is mainly designed to be able to analyze the voice of a user. This will be done through the use of a microphone that will be connected to the computer. The computer recognizes the microphone connected. The recorded audio is then passed through sound processing algorithms to obtain usable data that the computer can use.

The information is returned as data that is a representation of the audio file. This recorded audio is then passed on to be tested on the trained data. The trained data is a large sum of a audio files, composed of many different people acting out different emotions, specifically angry/sad, happy/excited and neutral.

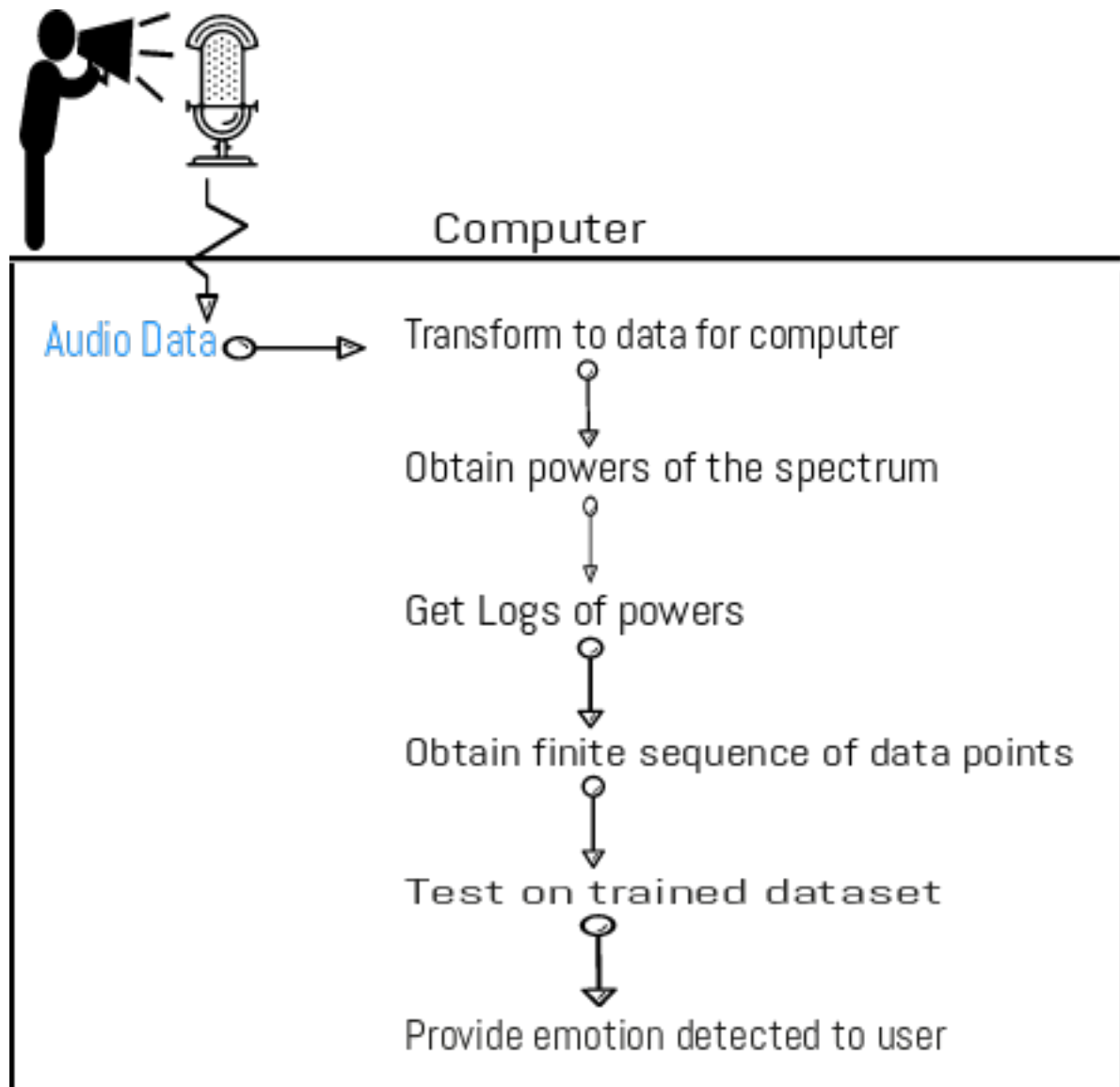


Figure 2: High Level Design

3.3 Low Level Design

This section focuses on the implementation of the system. The methods used and pseudo code for all functions that played a huge role in the implementation of the system.

In order for the system to accurately work, a large sum of audio files have to be trained, the audio files will be of the three emotions we are testing for. The audio will have to be turned

to data that the computer can understand and do calculations upon. To do this, Mel Frequency Cepstrum Coefficients(MFCC) will be used. MFCC is a feature extraction method for speech recognition, the goal of MFCC is to mimic the human hearing perception.

3.4 Feature Extraction

3.4.1 Mel Frequency Cepstrum Coefficient

The implementation of the MFCC used in the system was composed of four functions, Fast Fourier transform, Mel Scale, Log and Discrete Cosine Function. The use of these functions gave the feature that were used to be trained and to be tested.

These functions work in a linear method, each function depends on data from the previous method to accurately get the feature, The feature data given by the MFCC is a three dimension vector. The three dimensional vector can be manipulated to suit any format that the chosen classifier to be used to be able to understand the data. The audio file being processed by MFCC must be noise free as far as possible, as any other unnecessary noise on the audio might affect the accuracy of the testing.

The steps to follow when implementing MFCC:

1. Frame the signal given into short frames.
2. Calculate the periodogram estimate of the power spectrum
3. Apply the mel filterbank to the power spectra, sum the energy in each filter.

4. Take the logarithm of all filterbank energies.
5. Take the DCT of the log filterbank energies.

The process is shown below as a schematic diagram.

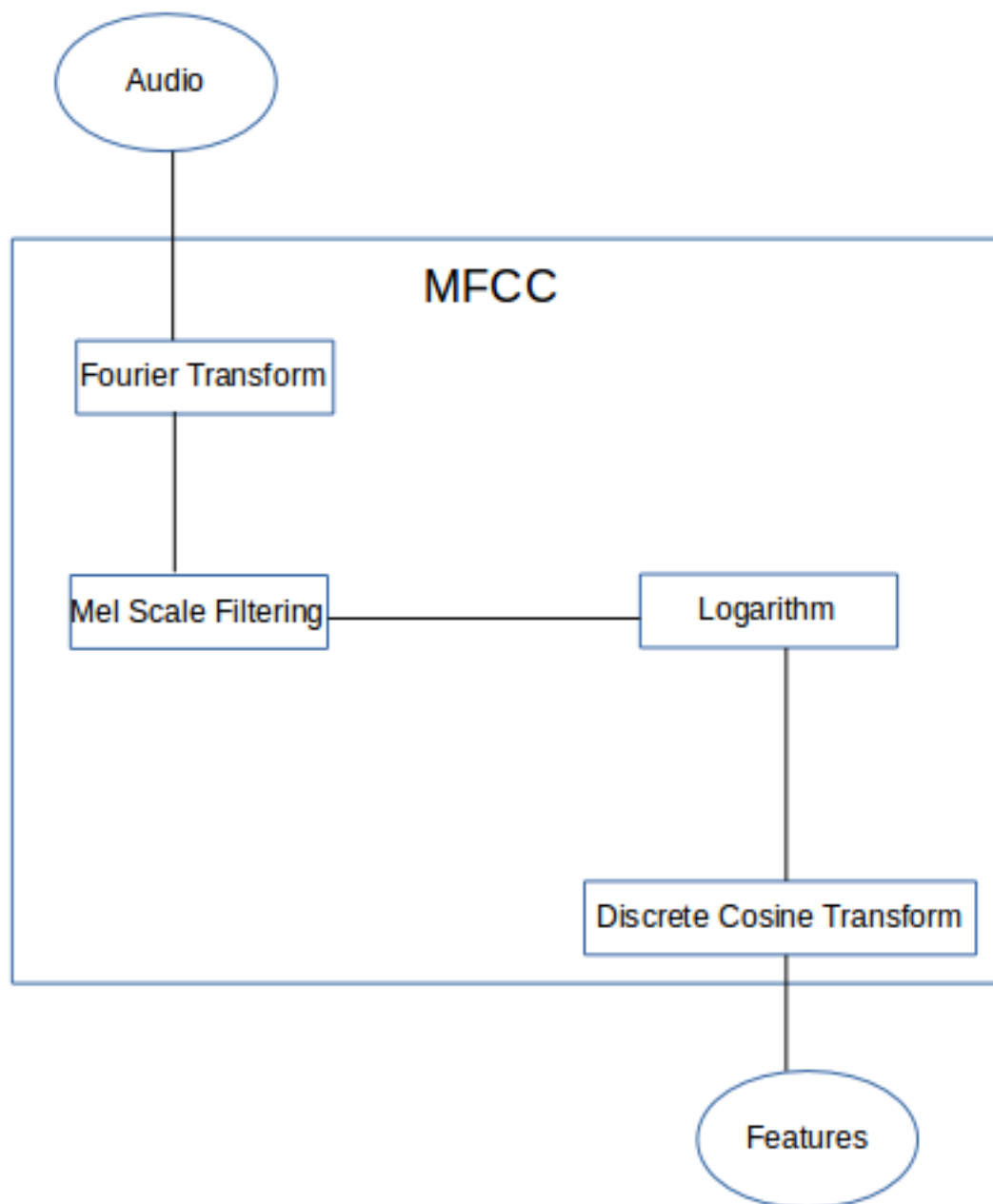


Figure 3: How Mel Frequency Cesptrum Coefficient works

3.4.1.1 Fast Fourier Transform(FFT)

The FFT is the first function that takes in the audio file and converts it to data that can be understood and processed by the computer. The data returned from this function is magnitude frequency response for each frame. The FFT algorithm does this by calculating the Discrete Fourier Transform(DFT). The DFT calculation is done as follows

$$c_{t,x}^l = \left| \sum f_j \exp\left[-i 2\pi \frac{jx}{N}\right] \right|, x \in (0, (N/2 - 1)) \quad (3.1)$$

FFT is not only limited to using DFT. DFT is one of the algorithms of FFT that require a large amount of computing power.

3.4.1.2 Mel Scale Filtering

This function is used to obtain the mel frequency spectrum. To do this, every element returned in the FFT function above will go through the algorithm for this function. The formula for doing this is:

$$m = 2595 \log_{10} \left(1 + \frac{c_{t,x}^l}{700} \right) \quad (3.2)$$

3.4.1.3 Logarithm

The next step is to get the log of the powers of all the frequencies. The aim of this is to mimic the human perception of loudness. The formula for doing this is:

$$c_{t,x}^1 = \log(m) \text{ where } x \in (1, N_m) \quad (3.3)$$

3.4.1.4 Discrete Cosine Function

After the logarithm has been taken. The next step is to remove

the speaker dependency by obtaining the Cepstral Coefficient. The formula for doing this is given as:

$$c_{t,x}^2 = c_{t,x}^1 \left[\frac{k(2j-1)\pi}{2N_d} \right] \text{ where } x \in (0, N_c < N_d) \quad (3.4)$$

All the above functions worked in finding the final data that was needed for classification.

3.5 Feature Classification

3.5.1 Support Vector Machines

In order for the system to work properly, the system will use SVM to train the data. Training data is a process of obtaining pre-existing data, which would be an audio set for the system. All the different emotions will be trained and labeled as a certain value during training. Happiness audio files could be labeled as 1, sadness 0 and neutral as 2.

but before any audio file is passed through the SVM, it has to be in understandable format for the SVM. In order to do so every audio file that comes to be classified has to pass the feature extraction phase.

Once this process is done the system will be able to test any file that satisfies the requirements required by system for testing.

Chapter 4

Code Documentation

4.1 Mel Frequency Cepstrum Co-efficients

4.1.1 Fast Fourier Transform

FFT is the first step of MFCC, it converts a signal from its original domain to a presentation in the frequency domain. The pseudo code can be seen below.

```
function FFT(sig):  
    len of sound data = len(sig)  
    If(Length of sound data <= 1):  
        Return sig  
    Even = FFT(All values that are even)  
    Odd = FFT(All values that are odd)  
    T = [exp(-2j*pi*k/N)*odd[K] for k in range((N/2) - 1)]  
    Arr = [even[k] + T[k] for k in range(N/2) ] + [even[k] -  
        T[k] for k in range (N/2)]  
    return Arr
```

4.1.2 MelScale Filtering

MelScale puts together the higher frequencies more closely. The pseudo code can be seen below

```
function melScale(Arr[]):
    for(i in length(Arr)):
        Arr[i] = 2595*ln(1 + Arr[i]/700)
    return Arr
```

4.1.3 Logarithm

This function tries to mimic the human perception of loudness. The pseudo code can be seen below

```
function Log(Arr[]):
    newArr[]
    for(i in length(Arr)):
        newArr[i] = log(Arr[i])
    return newArr
```

4.1.4 Discrete Cosine Transform

DCT reduces the speaker dependency by obtaining the cepstral co-efficients. The pseudo code can be seen below

```
function DCT(newArr[]):
    dctarray[]
    k = length of newArr
    For(i in k-1):
        for(j in k):
            for(j in k):
```

```

        hold = hold + newArr[j]*cos(k*(2*j-1)*pi/2*k)
    dctarray = hold
    hold = 0

return newArr

```

4.1.5 Vectorization

Vectorization prepares the data into a format of a 3D vector. This vector is the features required for training or testing. The pseudo code can be seen below.

```

function vectorize(dctarray):
    vecarray = zeros(length(dctarray) + 2)
    for(i = 1 in length(vecarray) - 1):
        vecarray[i] = dctarray[i - 1]
    delta[]
    k = 2
    for(i in length(dctarr)):
        delta[i] = fill[j] fill[i]
        j++
    populatedelta = zeros(length(delta) + 2)
    for(i = 1 in length(delta) - 1):
        populatedelta[i] = delta[i-1]
    deltatwo[]
    j = 2
    for( i in length(delta)):
        deltatwo[i] = populatedelta[j] - populatedelta[i]
        j++
    featVec[] [] []
    for(i in length(dctarray)):
        featVec[i][i][i] = dctarray[i], delta[i], deltatwo[i]
return featVec

```


4.2 Training

4.2.1 Support Vector Machine

The SVM is used to train on the sound dataset after the dataset has gone through MFCC and the features of the sound clips have been extracted, this will then create a model that will be used to predict the class of a certain audio clips' emotion. Happiness is labelled 1, anger as 2 and neutral as 0.

Every emotion should correspond to its label as follows

```
X = np.array([happy01,...,happy27,ang01,...,ang27,neutr01,...,neutr27])
Y = np.array([1,...,1,2,...,2,0,...,0])
import SVM
clf = svm.SVC()
clf.fit(X,Y) joblib.dump(clf, 'trained.pkl')
```

Once the pickle file is created, it can be used to classify other dataset sound clips, this is done as follows.

```
clf = joblib.load('trained.pkl')
clf.predict(VectorArray)
```

4.3 Libraries used

1. import scipy.io.wavfile as wav : Reads the audio file into audio data
2. from sklearn.svm import SVC : Calls the SVM which does the training
3. from sklearn.externals import joblib: Used to save and load the training file

Bibliography

- Kwon, O.-W., Chan, K., Hao, J., and Lee, T.-W. (2003). Emotion recognition by speech signals. Technical report, Institute for Neural Computation University of California, San Diego, USA.
- Rao, K. S., Kumar, T. P., Anusha, K., Leela, B., Bhavana, I., and Gowtham, S. V. (2003). Emotion recognition from speech. Technical report, School of Information Technology, Indian Institute of Technology Kharagpur.
- Yacoub, S., Simske, S., Lin, X., and Burns, J. (1990). Recognition of emotions in interactive voice response systems. Technical report, HP Laboratories, Palo Alto.