****

**UNIVERSITY OF NAIROBI**

**COLLEGE OF ARCHITECHTURE AND ENGINEERING**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING**

**AUTOMATIC GENDER RECOGNITION BY VOICE**

**PROJECT INDEX: PRJ 157**

**KONZI SAMUEL**

**F17/22951/2008**

**SUPERVISOR: MR. OSCAR ODENG**

**EXAMINER: PROF. VITALIS K. ODUOL / DR. A. NYETE**

Project report submitted in partial fulfilment of the requirement for the award of the Degree of: Bachelor of Science in Electrical and Information Engineering of the University of Nairobi

**Submitted on: 14th June 2017**

# DECLARATION OF ORIGINALITY

**NAME OF STUDENT:**

Konzi Samuel

**REGISTRATION NUMBER:**

F17/22951/2008

**COLLEGE:**

Architecture and Engineering

**FACULTY/ SCHOOL/ INSTITUTE:**

Engineering

**DEPARTMENT:**

Electrical and Information Engineering

**COURSE NAME:**

Bachelor of Science in Electrical & Electronic Engineering

**TITLE OF WORK:**

1. I understand what plagiarism is and I am aware of the university policy in this regard.
2. I declare that this final year project report is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people’s work or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi’s requirements.
3. I have not sought or used the services of any professional agencies to produce this work.
4. I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.
5. I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with University anti-plagiarism policy.

SIGNATURE: ………………………………………………………………………………………

DATE:

………………………………………………………………………………………

# CERTIFICATION

This report has been submitted to the Department of Electrical and Information Engineering with my approval as supervisor.

Mr. Oscar Odeng

Signature ……………………………..

Date … ……………………………..

# ACKNOWLEDGEMENT

I would like to thank my project supervisor Mr. Oscar Odeng who’s guidance and constructive criticism has been of great help in the throughout my project.

I also thank my project group members for the sharing of ideas that were helpful directly or indirectly in the project research and execution.

I would also like to thank my parents and sister for their support in ensuring I carried out my studies smoothly

Finally I thank the almighty God for help and the strength to work through this project.

# TABLE OF CONTENTS

[DECLARATION OF ORIGINALITY ii](#_Toc485227822)

[CERTIFICATION iii](#_Toc485227823)

[ACKNOWLEDGEMENT iv](#_Toc485227824)

[TABLE OF CONTENTS v](#_Toc485227825)

[LIST OF FIGURES viii](#_Toc485227826)

[LIST OF TABLES ix](#_Toc485227827)

[ABSTRACT x](#_Toc485227828)

[ABRREVIATIONS xi](#_Toc485227829)

[CHAPTER ONE: INTRODUCTION 1](#_Toc485227830)

[1.1 Why Gender Detection by speech 1](#_Toc485227831)

[1.2 Project Objectives 2](#_Toc485227832)

[1.3 Project scope 2](#_Toc485227833)

[1.4 Project organization 2](#_Toc485227834)

[CHAPTER TWO: LITERATURE REVIEW 3](#_Toc485227835)

[2.1 Introduction 3](#_Toc485227836)

[2.2 Nature of Speech 3](#_Toc485227837)

[2.3 Speech Signal Processing 3](#_Toc485227838)

[2.3.1 Quantization 4](#_Toc485227839)

[2.3.2 Fourier transform 4](#_Toc485227840)

[2.3.3 Discrete Cosine Transform 6](#_Toc485227841)

[2.4 Speech Enhancement 6](#_Toc485227842)

[2.4.1 Signal to Noise Ratio 6](#_Toc485227843)

[2.4.2 Spectral Noise Subtraction 7](#_Toc485227844)

[2.4.3 Voice Activity Detector 7](#_Toc485227845)

[2.5 Features of speech 7](#_Toc485227846)

[2.5.1 Pitch 8](#_Toc485227847)

[2.5.1.1 Pitch Detection by Autocorrelation 9](#_Toc485227848)

[2.5.2 Mel Frequency Cepstral Coefficients (MFCC) 10](#_Toc485227849)

[2.5.2.1 Framing 12](#_Toc485227850)

[2.5.3.2 Windowing 12](#_Toc485227851)

[2.5.3.3 Fast Fourier Transform (FFT) 13](#_Toc485227852)

[2.5.3.4 Mel frequency Filter Bank 13](#_Toc485227853)

[2.5.3.5 Cepstrum 13](#_Toc485227854)

[2.5.3 Acoustic Features of Speech 14](#_Toc485227855)

[2.6 Machine Learning Techniques 18](#_Toc485227856)

[2.6.1 Introduction 18](#_Toc485227857)

[2.6.1.1 Supervised learning 18](#_Toc485227858)

[2.6.1.2 Unsupervised learning 19](#_Toc485227859)

[2.6.2 Model Representation 19](#_Toc485227860)

[2.6.2.1 Cost Function 20](#_Toc485227861)

[2.6.2.2 Gradient Descent 20](#_Toc485227862)

[2.6.2.3 Feature Scaling 20](#_Toc485227863)

[2.6.3 Support Vector Machines 21](#_Toc485227864)

[CHAPTER THREE: SYSTEM DESIGN 25](#_Toc485227865)

[3.1 Introduction 25](#_Toc485227866)

[3.2 Feature Extraction 26](#_Toc485227867)

[3.2.1 Pitch Extraction 26](#_Toc485227868)

[3.4 Acoustic Features Model 28](#_Toc485227869)

[3.5 Pitch Based Model 30](#_Toc485227870)

[3.6 MFCC Based Model 31](#_Toc485227871)

[CHAPTER FOUR: RESULTS 32](#_Toc485227872)

[4.1 Acoustic Features Based Model 32](#_Toc485227873)

[4.2 Pitch Based Model 34](#_Toc485227874)

[4.3 MFCC Based Model 36](#_Toc485227875)

[4.4 Combined MFCC and Pitch Model 38](#_Toc485227876)

[CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS 40](#_Toc485227877)

[5.1 CONCLUSION 40](#_Toc485227878)

[5.2 RECOMMENDATIONS 40](#_Toc485227879)

[REFERENCES 41](#_Toc485227880)

[APPENDIX 42](#_Toc485227881)

[Acoustic model main code 42](#_Toc485227882)

[Pitch, MFCC, combined pitch and MFCC main code 43](#_Toc485227883)

# LIST OF FIGURES

[Fig 2.1: Conversion of an analog signal to digital signal [4] 4](#_Toc485230558)

[Fig 2.2 FFT Plot of a speech signal 5](#_Toc485230559)

[Fig 2.3: Block diagram of autocorrelation algorithm 10](#_Toc485230560)

[Fig 2.4: Block diagram of MFCC stages of processing 11](#_Toc485230561)

[Fig 2.5: Framing diagram (Four frames, with three overlaps) 12](#_Toc485230562)

[Fig 2.6: Mel- space Filter Bank (12 filters) 13](#_Toc485230563)

[Fig 2.7: Types of Machine Learning [10] 18](#_Toc485230564)

[Fig 2.8: Machine learning model representation [10] 20](#_Toc485230565)

[Fig 2.9: SVM decision Boundary, hyper plane and support vectors 21](#_Toc485230566)

[Fig 2.10: Different values of regularization parameter (C) 24](#_Toc485230567)

[Fig 3.1: Flow chart for an ADG 25](#_Toc485230568)

[Fig 3.2: Plot of mean fundamental frequency vs number of features (main feature) 29](#_Toc485230569)

[Fig 3.3: Plot of skew vs number of samples (not main feature) 29](#_Toc485230570)

[Fig 3.4: Plot of pitch vs number of samples (ELSDSR train dataset) 30](#_Toc485230571)

# LIST OF TABLES

[Table 1: Train accuracy of the 20 acoustic Features for c=1, g=0.1 32](#_Toc485231857)

[Table 2: Test accuracies for the 20 acoustic Features c=0.1, g=0.1 33](#_Toc485231858)

[Table3: Test accuracies for the 20 acoustic Features c=1, g=0.1 33](#_Toc485231859)

[Table 4: Combination of different selected features. 34](#_Toc485231860)

[Table 5: Test on different parameter values of c and g on the main features. 34](#_Toc485231861)

[Table 6: Train accuracies of pitch model using ELSDSR dataset 35](#_Toc485231862)

[Table 7: Test model based on Pitch. 35](#_Toc485231863)

[Table 8: Performance of pitch for a smaller training set. 36](#_Toc485231864)

[Table 9: Performance of pitch when tested on raw recorded files 36](#_Toc485231865)

[Table 10: Train model based on MFCC. 37](#_Toc485231866)

[Table 11: Test model based on MFCC. 37](#_Toc485231867)

[Table 12: Test model based on MFCC (smaller training set). 37](#_Toc485231868)

[Table 13: Performance of MFCC model when tested on raw recorded files. 38](#_Toc485231869)

[Table 14: Train model based on MFCC and pitch. 38](#_Toc485231870)

[Table 15: Test model based on MFCC and pitch 38](#_Toc485231871)

[Table 16: Test model based on MFCC and pitch (smaller training set) 39](#_Toc485231872)

[Table 17: Test model based on MFCC and pitch on raw recorded files 39](#_Toc485231873)

# ABSTRACT

The aim of this project is to develop an automatic gender detection system by voice. Gender detection is a subset of speech recognition, and is incorporated in systems that used speech recognition. In the recent years there has been need for extra biometric security measures for example as well as a rise in use of telecommunication devices. This has made the practicability of speech recognition and automatic gender detection to increase in terms of application. This means great research is going into developing such systems, and the main objective is to develop systems with higher accuracy.

Speech exhibits special features useful in gender detection. These features are then given to a machine learning algorithm which learns from the data and develops a model based on the input data given to develop a suitable model as required. In this project pitch, the biggest discriminator is used as well as Mel frequency cepstral coefficients. Support vector machines, a powerful machine learning algorithm is applied to develop a classification model. In this project several models are developed and tested in an attempt to identify key features and design specification for automatic gender detection.

# ABRREVIATIONS

AGD - Automatic Gender Detection

C -Regularization parameter

CMN -Cepstral Mean Normalization

DARD -Data Reduction Methods

DCT - Discrete Cosine Transform

DFT - Discrete Fourier Transform

ELSDSR- English Language Speech Database for Speaker Recognition

FFT - Fast Fourier Transform

g -Sigma (Support Vector Machine Parameter)

LIBSVM-Support Vector Machines Library

LPC -Linear Predictive Coding

LPCC - Linear Predictive Cepstral Coefficients

MFCC - Mel Frequency Cepstral Coefficients

PLP -Perpetual Linear Prediction

RASTA-PLP -Relative Spectra filtering of log domain coefficients

SVM - Support Vector Machines

SNR - Signal to Noise Ratio

VAD -Voice Activity Detector

RBF -Radial Basis Function Kernel (Gaussian kernel)

# 

# CHAPTER ONE: INTRODUCTION

## 1.1 Why Gender Detection by speech

In the last few years technological growth has seen computers being part of our day today lives. Personalized human-machine interaction is now becoming popular with great research being done in making it more specific and as natural just like human-human communication. Speech recognition systems has enabled people to talk to their phones, for example, instead of typing the various phone related tasks.

Machine learning on the other hand has been a thriving field in the recent years. Evolving from artificial intelligence in which software is automated to understand human natural traits and thus facilitating scientific research. Machine learning according to [1] is the field of study that gives computers the ability to learn without explicitly being programmed. It involves various fields such as face recognition, finger print, and emotion study from gait and so on. In this project the main focus is on gender recognition by speech.

Automatic gender recognition has a variety of applications that involve speaker recognition systems, speech analytics and human machine interaction. It can be applied in sorting telephone calls by gender for gender sensitive surveys, identifying the gender and removing the gender specific components thus giving enhanced bandwidth. [2]

AGD involves preprocessing and feature extraction of voice data, and then classification of these features by use of a known training dataset. Various speech processing techniques are preferred as well as classification algorithms. AGD proves to be a difficult task looking at the percentage accuracies obtained in past studies. It works based on the premise that speech will exhibit enough characteristics to enable gender classification, however variation in speech signals has sometimes shown some character closely related to one gender yet the person is of the other gender. An ideal model should be able to give good performance even under real life environment with conditions such as noisy speech, silence which is quite difficult to achieve.

According to the human ear, pitch is the biggest discriminator on making a decision on gender. In terms of pitch location, absolute frequency has been found to be the most important information for gender classification. Cepstral features like Mel Frequency Cepstral Coefficients (MFCC) are also used in classification since they provide unique features important also for gender classification. A combination of both features is found to give the best results and thus this project is based on a model that incorporates both features.

Digital signal processing is applied on the speech for feature extraction. It involves speech enhancement in for example reducing background noise thus obtaining a big enough signal to noise ratio. Support vector machines (SVM), a popular machine learning algorithm is applied to perform the classification task. The various concepts in speech processing and classification are explained in this project.

## 1.2 Project Objectives

Voice based gender detection using machine learning.

## 1.3 Project scope

Identify useful features for gender classification.

Extract some of the speech features from wav files for gender classification.

Perform real time gender classification based on voice.

## 1.4 Project organization

Chapter one is an introduction and the objectives of the project

Chapter two gives a review of the techniques used in digital speech processing and machine learning.

Chapter three is a description of the implementation of the AGD in MATLAB and the various methods employed

Chapter four shows the results obtained and the various waveforms obtained in speech processing.

Chapter five is a conclusion of the project and recommendations for future research.

Appendices contains the main code used in the project implementation.

# CHAPTER TWO: LITERATURE REVIEW

# 2.1 Introduction

This chapter a review of the various speech characteristics, digital speech processing methods and machine learning classification methods is presented. In view of the objectives of this project the literature has been divided into the following sections:

1. Nature of speech
2. Speech signal processing
3. Speech enhancement
4. Features of speech
5. Machine learning classification techniques

## 2.2 Nature of Speech

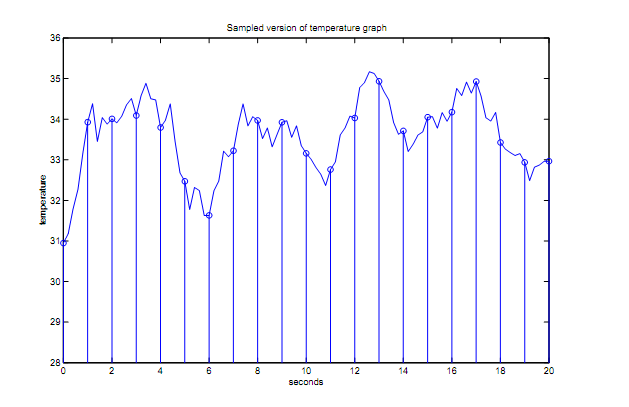
Speech is a vibration produced by the vocal folds and has uniqueness effected by the tongue, lips and jaws. The human speech ranges from 50Hz to 3400 kHz and in terms of hearing the frequencies range in between 20Hz to 20 KHz.

Fundamental frequency (pitch: discussed later in detail) is the most obvious voice characteristic to the human ear in identifying gender. Adult male fundamental frequencies range from 85Hz -180Hz while in female the range is 165Hz- 225Hz [3]. Speech signals are analog by nature and to study its features using computers, they need to be converted to a digital signal.

## 2.3 Speech Signal Processing

As mentioned speech signal needs to be converted to a digital signal for processing. The conversion is done by a process known as sampling. Sampling converts speech, a continuous time signal to a discrete time signal. This is done at intervals of choice known as the sampling interval, which is the period *T* and thus the reciprocal of frequency. In sampling some data is lost, and to avoid losing important information of the speech the sampling rate needs to be high. According to the sampling theorem, the minimum sampling rate is twice the frequency of the highest frequency contained in the signal source [1].

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

represents the Bandwidth of the signal which is also the highest frequency contained in the signal.

##### Fig 2.1: Conversion of an analog signal to digital signal [4]

### 2.3.1 Quantization

Quantization is signal compression of continuous signals to smaller sets of discrete values. It is done for either transmission over channels or storage purposes because of bandwidth limitation [1]. The conversion is meant to be accurate and errors arising from the quantized signal with respect to the sampled signal is called quantization error. The quantization error is reduced by setting the quantization level to higher bits. Compact disks are usually sampled at 44.1 KHz. The quantization level is set at 16 bits and thus the amplitude of the signal gives 216 possible values per sample or 65,536 discrete levels [5].

### 2.3.2 Fourier transform

Fourier analysis is the study of signals and systems using sinusoidal representations. This is after Joseph Fourier (1768-1830) who contributed to theory of representing functions as weighted superposition of sinusoids. This is useful in deconstruction of a signal into its components for ease of study.

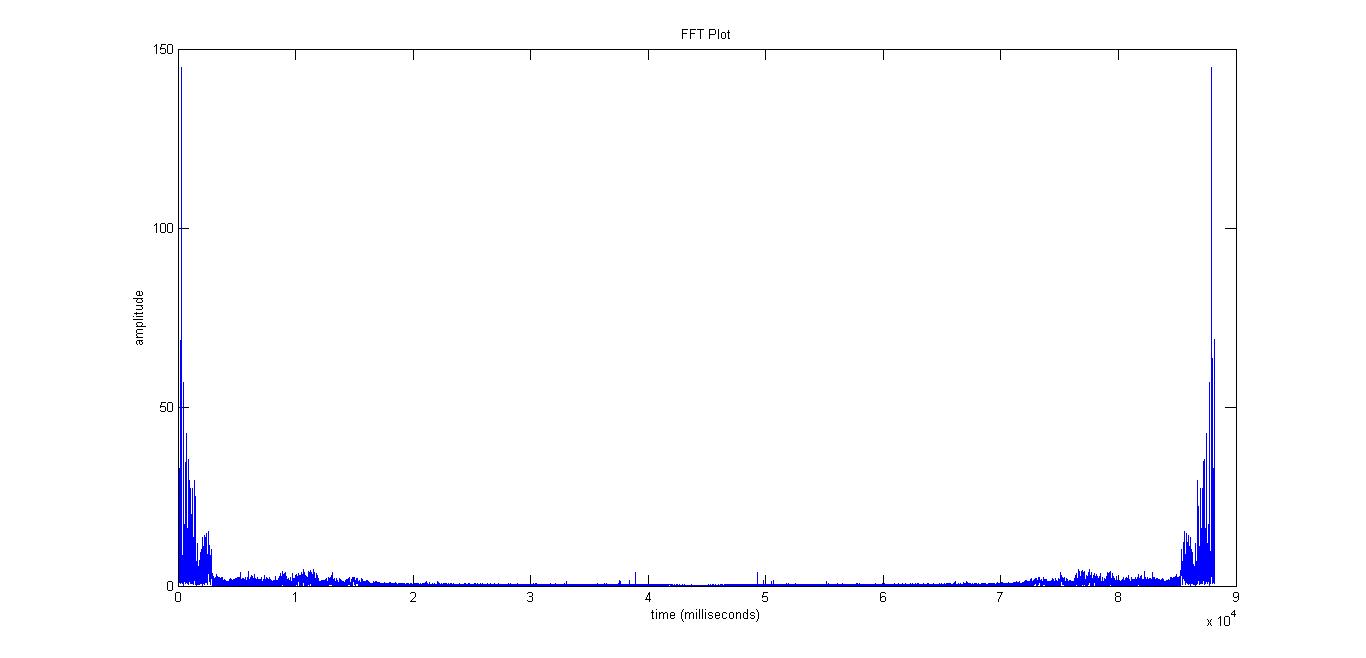
For a continuous function of time with input and

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Where   
 Fourier methods include:

1. Discrete time Fourier transform
2. Discrete Fourier transform
3. Continuous time Fourier transform
4. Continuous time transform

For a Discrete Fourier Transform to be obtained, the frequency domain of the DTFT is sampled.



##### Fig 2.2 FFT Plot of a speech signal

Out of the Fourier transform methods, speech is mostly analyzed by DFT because of the nature of its values. Speech is mostly implemented using the FFT algorithm. This is because it reduces the complexity of computing the DFT from order () to order [1].

### 2.3.3 Discrete Cosine Transform

DCT is similar to DFT but different in that the transformation of its signal from time to frequency domain is through representing the output as a sum of cosine functions all with different frequencies. DCT has numerous applications in image and video processing. The higher frequencies that exist in sound signals are discarded and thus most of the frequencies which are low (of the transformed signal) are kept.

The DCT is defined by;

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

## 2.4 Speech Enhancement

Speech recognition systems may not be operated in a laboratory environment and thus in recognition in the real-world comes a challenge of noise or even silence. Speech enhancement is of importance in improvement of quality of speech and obtaining as much initial speech features even in a noisy environment.

### 2.4.1 Signal to Noise Ratio

The signal to noise ratio is the measure of the signal strength with respect to the noise strength. The noise is characterized by its standard deviation *Sn.* For a signal lying between the boundaries , the SNR is given by:

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

In calculating a direct proportion without using logarithmic terms,

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Where signal and

A larger signal to noise ratio is preferred since the wanted signal can be easily obtained and the noise eliminated.

### 2.4.2 Spectral Noise Subtraction

Speech noise subtraction assumes existence of background noise in the signal being considered. Assuming a speech signal in a noisy environment, give the result of the signal obtained as:

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Where is the resultant signal and the noise introduced by,

### 2.4.3 Voice Activity Detector

VADs evaluate the existence of silence or speech in a speech signal. Silence is common especially when someone is talking with pauses. Silence in most times is not interpreted by the machine correctly and thus removing the silence in signals is important so that the correct features can be obtained from a signal. They are applied in coders for bandwidth conservation. When there is silence, the speech coder stream is discontinued [1]. In speech recognition systems the same principle is applied only that the signal might not be discontinued during silence as a part of deep learning of the speakers behavior in speaking in a bid to strengthen recognition accuracy [6].

## 2.5 Features of speech

In AGD speech features have to be obtained for training and testing of the system. Pitch is an obvious consideration in this case since it and is included in systems and its various extraction features applied. Acoustic features however are applied since pitch alone isn’t an accurate discriminator. Several speech extraction methods that have been used in past research include;

* Mel Frequency Cepstral Coefficients(MFCC)
* Linear Predictive Coding(LPC)
* Linear Prediction Cepstral Coefficients(LPCC)
* Perceptual Linear Prediction(PLP)
* Relative Spectra Filtering of log domain coefficients PLP(RASTA-PLP)

LPC and LPCC are in the current technology considered low end in terms of resources required, implementing and doesn’t work best in multi-speaker testing in which our project is about. RASTA-PLP on the other hard is quite hard to implement and requires a great deal of resources to incorporate it in a system. In this project therefore, only MFCC which is the most popularly used will be considered as it relatively easy to implement and can be applied as required in a multi speaker case.

### 2.5.1 Pitch

This is the most obvious feature that can be obtained from speech. It is defined as the fundamental frequency of vibration. Pitch period on the other hand according to [2] is the interval between two consecutive voiced excitation cycles. [1] Spectral envelops of voiced sounds exhibit pitch and formants for a sampling rate of 8 KHz. For unvoiced sounds, no pitch or formant characteristics are revealed. A threshold can be set for as system in gender recognition such that frequencies above the threshold represent female classification and those below represent male. However using pitch alone is only possible for non-noisy signals and is made more difficult by the overlap of the boundary between male and female frequency range. Noise often buries real harmonic peaks and creates false peaks which cause problems to find pitch.

Speech signal is a slowly time varying signal (quasi-periodic) and thus special algorithms are needed to estimate pitch and its feature to give enough information for gender detection. This is mostly done in time domain accurately as it deals with the speech waveform directly. In frequency domain, the transform process, may lead to loss of some features useful for finding pitch. Several methods are used for operations on the waveform.

Time Domain Methods

* Short Term Auto correlation Function
* Average Magnitude Difference Function
* Zero crossing Rate
* Data reduction methods(DARD)
* Parallel processing

Frequency Domain Methods

* Harmonic product spectrum
* Harmonic to sub harmonic ratio
* Spectral Autocorrelation
* Comb Transformation

In this project autocorrelation method is applied in pitch extraction and the various steps involved in extraction using this method are discussed.

#### 2.5.1.1 Pitch Detection by Autocorrelation

In autocorrelation the correlation of a signal is measured with a copy of itself. When main peak for this function occurs it is detected and is saved as the pitch period. Center clipping and infinite clipping is applied. Center clipping is useful in eliminating multiplication required for autocorrelation. For a given discrete time signal x (n) defined for all values of n, the autocorrelation function is generally defined as

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Where is the delay, N is the total number of samples in a short frame [7].

Since speech is s quasi periodic signal, it is divided into frames and short time autocorrelation function applied for each frames and the average pitch is obtained from all the frames. The figure 2.3 is a block diagram for the algorithm.

The speech is passed through a 1-900Hz low pass filter.

The clipping threshold is computed, for a clipping level of 0.68 of the smaller peak of the absolute sample values [7]. After determining the clipping level, the speech section is center clipped and then autocorrelation performed on all the frames. To determine voiced and unvoiced segments, autocorrelation is computed at zero delay.

The maximum values of the autocorrelation function obtained and a threshold set for voice and unvoiced segment.

Speech signal

LPF (0-900Hz)

Calculate clipping level CL

Center clipping (c) and infinite peak clipping (v)

Energy of center clipped signal

Autocorrelation Function

Compare peak value with set threshold

Less: pitch=0 (unvoiced) or pitch= 1/index (voiced)

Pitch

##### Fig 2.3: Block diagram of autocorrelation algorithm

### 2.5.2 Mel Frequency Cepstral Coefficients (MFCC)

Mel frequency analysis is based on human ear perfection of frequencies. It acts as a filter, concentrating only certain frequency components [8]. MFCCs represent the short term power spectrum of sound and are based on the linear cosine transform of a log power spectrum on a linear Mel scale of frequency. They are essentially obtained from the Mel-spectrum.

*Speech signal*

Frame blocking

*Frames of speech*

Windowing

FFT

*Spectrum*

Mel Filter bank

*Mel spectrum*

Cepstrum

*Mel cepstrum*

##### Fig 2.4: Block diagram of MFCC stages of processing

For an input signal, with limit value;

Then the, log of the Mel spectrum;

Cepstral analysis is performed on log;

Taking the FFT;

The resulting overall equation is given by;

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

The steps involved in MFFC feature extraction areas shown in the block diagram below

#### 2.5.2.1 Framing

This is the diving of the continuous speech signal to frames to achieve stationarity. This means working with windows of 20-30 ms and N samples. For a long frame, change in signal properties may be too much in one window. For a too short frame, narrow-band components would be compromised. However a good choice of samples in each frame is 256 and the number of frames is given by

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

The edges of the signal add harmonics and windowing is used to tone down the edges. Overlaps are used with the first frame overlapping the second as shown in the figure:

****

##### Fig 2.5: Framing diagram (Four frames, with three overlaps)

Overlap time is the time from a new frame until the end of the current frame. Deframing is now the combining of the individual frames into one speech signal. The frame usually have discontinuities at the beginning and end of frame because of overlapping.

#### 2.5.3.2 Windowing

Spectral distortion is reduced by the use of windows so that the signal is zero at the beginning and at the end. For N number of samples per frame defined by the window [, the resultant signal is given by.

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

The most used window functions are Bartlett, Rectangular, Hanning, Hamming and Backman. The hamming window introduces the least distortion to a signal and thus is the most commonly used. Its equation is given as

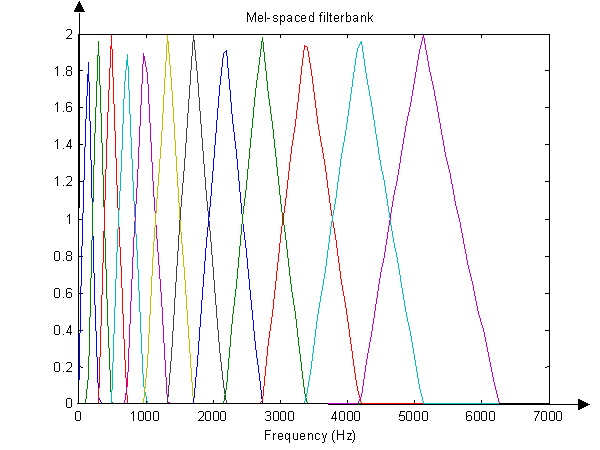
|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

#### 2.5.3.3 Fast Fourier Transform (FFT)

As explained in section 2.3 in speech signal processing, the FFT is a fast algorithm to implement DFT defined by samples N. The time domain samples of each frame are converted into frequency domain. For each short analysis window from the windowing stage, a period gram estimate of the power spectrum is obtained.

#### 2.5.3.4 Mel frequency Filter Bank

Consists of filters that are non-uniformly spaced on the frequency axis with more filters on the low frequency side and less on the high frequency. It is based on the human perception of frequency contents of speech.



##### Fig 2.6: Mel- space Filter Bank (12 filters)

#### 2.5.3.5 Cepstrum

Cepstrum is the inverse Fourier transform of the logarithm of the power spectrum of a signal.

The logarithmic Mel spectrum is converted to time to obtain MFCCs. This is done by applying discrete cosine transform. Speech is thus represented as a sequence of cepstral vectors, a satisfactory representation of the signal properties. The vectors are the coefficients that are fed to the classier for automatic gender detection purposes.

### 2.5.3 Acoustic Features of Speech

Acoustic features are useful in gender identification. In this project a set of 20 features from kaggle [9]are used.

**Mean Frequency (KHz) -** is the weighted average of frequency by amplitude. Instead of each data points contributing equally, some amplitude contribute more to the mean frequency. Mean on the other hand as in traditional statistics is defined as the addition of all the frequency samples and diving by the number of samples.

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Where is the actual data frequency samples and is the total number of the frequency samples.

For weighted average we have the equation;

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Where is the amplitude measure and is the weight assigned to each amplitude.

* **Standard Deviation-**This is a measure of how a signal fluctuates from the weighted mean. Variance on the other hand gives the power of this variation and thus standard deviation is the square root of variance.

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Where and

For a noisy signal, the mean describes the signal strength while the standard deviation describes the noise and thus the Signal to Noise Ratio (SNR) can be obtained by

* **Median Frequency-** This is the median normalized frequency of a signal or the particular frequency that would divide the power spectrum into two parts of equal area. For a data with even set of values, the data is arranged in ascending order, the two middle numbers picked and divided by 2. For odd set of data it is simply the middle number for data arranged in ascending order.
* **First Quartile (Q25) -** Represents the median frequency of the lower half of a dataset. A quarter of the values in the dataset are below the first quartile and three quarters lie above this value (KHZ).
* **Third Quartile (Q75) -** Represents the median frequency of the upper half of a dataset. Three quarters of the values in the dataset are below the third quartile and a quarter lie above this value.(KHz)
* **Interquartile Frequency Range-** Is also known as the mean spread and is the first quartile subtracted from the third quartile. It is important in finding outliers in a dataset.
* **Skewness-** is the measure of the degree of asymmetry in a set of statistical data. It’s obtained by getting the difference between median from the mode and then dividing by the standard deviation.

It is computed using the formula.

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

For,

When the spectrum is skewed to the left

When the spectrum is symmetric

When the spectrum is skewed to the right.

Spectrum asymmetry increases with.

* **Kurtosis-**Describes the distribution of observed data around the mean. It is calculated using the equation;

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

For,

When the spectrum is platikurtic, (data with fewer items at the center and at the tails than the normal curve but has more items in the shoulders)

When spectrum shows a normal shape

When the spectrum is leptokurtic (has more data items at the center and at the tails, with fewer items at the shoulder relative to normal distribution with the same mean and variance)

* **Mean dominant Frequency-**is average of the frequency that shows the maximum energy among all the frequencies in a spectrum.
* **Minimum Dominant Frequency-**is the minimum of the dominant frequency measured across the acoustic signal
* **Maximum Dominant Frequency-**is the maximum of the dominant frequency measure across the acoustic signal
* **Dominant Frequency range-**range of dominant frequency measured across the acoustic signal.
* **Spectral Entropy-**This is describes as the complexity of a system.

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Where relative amplitude of the frequency

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Number of frequencies

For a noisy signal spectra entropy tends toward 1and tends to zero for a clean signal

* **Spectral Flatness -** is measure used to characterize an audio spectrum. Provide a way of quantifying noise as a sound and not as tone and is calculated as the ratio between the geometric mean and the arithmetic mean

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Where relative amplitude of the frequency

Number of frequencies

For a noisy signal, the *sfm* tends towards 1and for a clean signal it tends to 0.

Has no units and returns a value 1 or 0.

* **Frequency centroid-** Is the midpoint frequency of the measured occupied bandwidth as calculated by the occupied bandwidth marker. It’s calculated according to the formula.

|  |  |  |
| --- | --- | --- |
|  |  | (2. ) |

Where Frequencies

Is the relative amplitude of the frequency

Is the number of frequencies

* **Peak frequency-**Refers to the frequency with the highest energy. (Not included in the test dataset due to cpu time in calculating the value)
* **Mean Fundamental Frequency-** It is the mean of all the fundamental frequency values across the length of speech signal.

For a periodic signal like speech it is defines as the lowest frequency. For speech which is a quasi-periodic in nature, only short time analysis give or time domain analysis gives the fundamental frequency.

* **Minimum Fundamental Frequency-**is the minimum frequency having a peak among all the frequencies in the spectrum of an acoustic signal.
* **Maximum Fundamental Frequency-**is the maximum frequency having a peak among all the frequencies in the spectrum of an acoustic signal
* **Modulation index-**is calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range.

## 2.6 Machine Learning Techniques

### 2.6.1 Introduction

Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed [10]. This means that the computer learns from experience of handling different related tasks and its performance on testing shows an improvement in learning. It is closely related to deep learning in which most of the tasks given to the computer are intuitive tasks that for human beings are easy to solve. The tasks are presented to it in relation to simple ones i.e. building complicated concepts from simpler ones. A graph of how these concepts are built on top of each other is a deep graph with layers and hence the term deep learning [11].

Machine learning algorithms are divided into;

* Supervised learning
* Unsupervised learning
* Reinforcement learning

Machine Learning

Supervised

Reinforcement

Unsupervised

##### Fig 2.7: Types of Machine Learning [10]

#### 2.6.1.1 Supervised learning

It involves instances used by machine learning algorithm that are labelled and thus a mapping function based on the input data and the known labels (corresponding positive output ) is learnt. The learning is supervise in that the algorithm makes predictions on the training data iteratively and halts when the algorithm achieves the accepted performance level.

Supervised learning is further divided into;

* Classification – is a problem that poses an output that is discrete or can be categorized for example ‘male’ or ‘female’, ‘zero’ or ‘one’ as in binary classes and so on.
* Regression – is a problem that poses a continuous output that has a value such as ‘length’, ‘weight’ and so on.

Examples of supervised learning algorithms include;

* Support vector machines –used for classification problems
* Linear regression – used for regression problems
* Logistic regression
* Naïve Bayes

#### 2.6.1.2 Unsupervised learning

Unsupervised learning deals with problems where the output of the data being considered is not known. The goal of these algorithms is to obtain a structured model of the input data by its on devises. This structure could be inform of forming clusters of related data or associative problems in which certain rules apply for examples people that watched a topic x are also interested with a topic y.

Examples of unsupervised learning algorithms include;

* K-means which is a clustering algorithm
* Gaussian mixture models
* Hidden Markov models

### 2.6.2 Model Representation

There are many different machine learning algorithms and each of them has different implementation and tricks for the different optimization hypotheses. In general to obtain high accuracies in learning algorithms, several practices need to be put in place for example using a large number of training examples. This avoids over fitting of data. Use of a small set of features is recommended, and or decreasing the regularization parameter (discussed later in detail) to obtain minimal error. An algorithm may perform well for the data trained on it but be inaccurate due to over fitting.

Considering a classification problem that outputs 0/1 for an input X

Training set

Learning Algorithm

Input (X)

Output (0/1)

Hypothesis

##### Fig 2.8: Machine learning model representation [10]

In running a machine learning algorithm, a grasp of the following terms is of great importance.

#### 2.6.2.1 Cost Function

The cost function in machine learning should be minimized. Considering a linearly separable dataset, a hypothesis objective would be to obtain the best line of fit for the data. By obtaining the best fit, the cost function is minimized as required. The cost function is denoted by J.

#### 2.6.2.2 Gradient Descent

Gradient descent is an optimization algorithm whose objective is to find values of the function to minimize the cost function. The intuition is that the parameters required for this minimization can’t be calculated and thus through gradient descent they are searched. After every iteration the parameters are updated and so on until gradient descent goes to the local minimum (convergence).

#### 2.6.2.3 Feature Scaling

In feature parameters, scaling of data is important in order to have a properly working gradient descent. A wide range of feature values would lead to inefficient oscillation when descending to the optimum values. Examples of good choice of ranges is

In feature normalization the formula used is given by;

|  |  |  |
| --- | --- | --- |
|  |  | (2. 21) |

Where µ is the mean average of values, is the range or the standard deviation and is the th feature value

### 2.6.3 Support Vector Machines

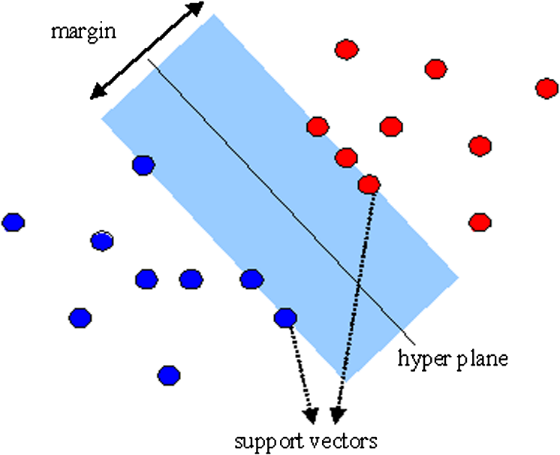
The features obtained from speech are given to a machine learning classifier to make a decision. In this project support vector machines (SVM) is used. The SVM algorithm is expected to give a model that is based on the training data feature and labels such that it predicts correctly values of the labels of test data, given only the test data features.

SVM is a supervised machine learning algorithm that tries to find decision boundaries of data based on maximum margin boundary [12]. It is thus also referred to as a maximum margin classifier. The plane separating the positive and negative values of a dataset in SVM is called a hyper plane, and is bordered by the large SVM margin. The data sets that lie on the margin separator are known as support vectors and they determine the margin which is the separator. The margin is the perpendicular distance between the decision boundary and support vectors.

For a two- class classification using the linear model, we have;

|  |  |  |
| --- | --- | --- |
|  |  | (2.22) |

Where is a fixed feature transformation, and the bias parameter is made explicit.



##### Fig 2.9: SVM decision Boundary, hyper plane and support vectors

Deriving the SVM optimization problem from logistic regression (which is easier to implement for machine learning problems), SVM requires the solution of the optimization case given by;

|  |  |  |
| --- | --- | --- |
|  |  | (2.23) |

Since m in this case is a constant, minimizing the function is independent of m, so we do away with it. We set so that for our new case the regularization term is governed by C and we get the equation.

|  |  |  |
| --- | --- | --- |
|  |  | (2.24) |

The tern in square brackets represents the cost from the training set and the second term represents the regularization term which is now governed by C, the regularization parameter.

The solution of the SVM optimization problems gives;

For training vectors, the SVM finds a separating hyper plane with maximal margin. In non-linear classification, an entity called kernels is introduced in SVM. Kernel functions give similarity between two vectors in a dataset. Kernels avoid working explicitly in feature space and thus vectors are mapped to a higher dimensional space. Kernel functions vary in implementation depending on the choice of kernel made.

The various SVM kernels include;

* Linear - is also referred to the no kernel method and is defined by;

|  |  |  |
| --- | --- | --- |
|  |  | (2.25) |

* Polynomial – is defined by the function

|  |  |  |
| --- | --- | --- |
|  |  | (2. 26) |

Where x and y are the input vectors and c is the free tradeoff parameter influencing the higher order versus lower order polynomials.

* Sigmoid – is defined by the function

|  |  |  |
| --- | --- | --- |
|  |  | (2. 27) |

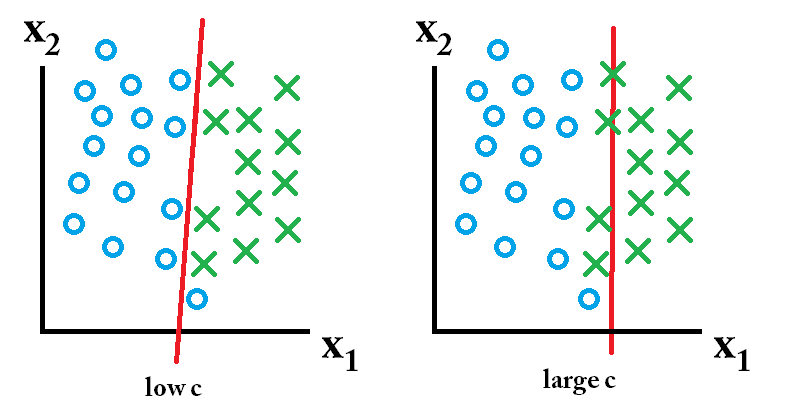
* Radial Basis Function (RBF) – is also referred to as the Gaussian kernel and is the most commonly used kernel. its similarity function is given by;

|  |  |  |
| --- | --- | --- |
|  |  | (2.28) |

Where are the selected landmarks for feature x for which similarity is being investigated. is refered to the squared Euclidian distance between the two feature vectors, and is a parameter and its relevance can be found by defining.

The Choice of kernel depends on user and the nature of the data being classified, but generally the RBF kernel is a good choice since for small values of the regularization parameter, it behaves like the linear kernel. However there are some rules of thumb to inform the decision; for a large number of features relative to training examples, the linear kernel is preferred. For cases of small number of training features and intermediate number of training examples, the Gaussian (RBF) kernel is a better choice.

Not all similarity functions make valid kernels, for a valid one the SVM package optimizations have to run correctly and not diverge. SVM is characterized by the parameters C (regularization parameter) and sigma ( (an RBF parameter) which are carefully chosen for best results. In typical designs large values of sigma, the features vary more smoothly but have higher bias and lower variance and the opposite is true for small values of sigma. On the other hand, for large values of C we obtain a lower bias and high variance. The figure below shows a hyper plane for different values of C.



##### Fig 2.10: Different values of regularization parameter (C)

# CHAPTER THREE: SYSTEM DESIGN

## 3.1 Introduction

The various design stages and methods described in the previous chapters are now implemented in this section. The design is made in MATLAB as it has many inbuilt functions and libraries that make the system design easy to work with. Various speech features were combined and used to develop an AGD model.

START

Signal Processing

Feature Extraction

Classification

Is result =1?

No

Result=female

Yes

Result=male

STOP

##### Fig 3.1: Flow chart for an ADG

. The flow chart above shows the general design stages for an AGD in their order of execution.

## 3.2 Feature Extraction

### 3.2.1 Pitch Extraction

In chapter 2 pitch by autocorrelation is described. The various stages were implemented in MATLAB to develop a real time pitch detection algorithm. The figure 2.11 is a flow chart for the pitch algorithm program flow. The program is made up of two files, the autocorrelation function and the pitch function.

* **Autocorrelation function**

The autocorrelation function has the inputs length of frame, sampling frequency and the frame of the segmented signal. It performs the following functions;

1. The signal is filtered using a 0-900Hz low pass filter.
2. Identify the clipping threshold by taking the absolute peak value of the first and last 1/3 of the frame.
3. Set the clipping level at 0.68 multiplied by the smaller of the peaks from the step 2 above.
4. Center clipping is performed with reference to the clipping level.
5. The energy of the clipped waveform is obtained
6. For each frame the maximum and minimum autocorrelation range is found and the indices for these ranges obtained, which when divided by the sampling frequency gives the pitch per frame.

* **Pitch Function**

Has the inputs of the speech signal and its sampling frequency. The speech is segmented into frames with overlaps and then the autocorrelation function is called. Median filtering is performed to eliminate noise from the speech signal by removing outlier values. The average of the pitch values for all the frames is calculate to obtain the final value.

**3.3 Classification using SVM**

As mentioned in chapter 2, SVM is a large margin classifier. The linear and RBF (Gaussian) kernels are used in this project to obtain a decision boundary.

The library LIBSVM [13] is a support vector machines toolbox and was used for the training and testing purposes. Training is done using the function for which we have an input of the training instance matrix and training label vector. The instance matrix is a matrix with m training instance and n features. The label vector is a matrix consisting of either 1 or 0 for our case; 1 represents male and 0 female. Inputs also includes LIBSVM options for example regularization parameter c, free parameters sigma etc. Its output is a model based on the kernel, parameters chosen, training instances and labels.

Predicting is done using the function and has the inputs, testing label vector, testing instance vector and the model obtained by running the the outputs consist of the predicted labels, accuracy in prediction and the decision values.

The problem in this project is a C-SVM classification, and the training vectors are of two classes with label. The algorithm solves the primal optimization problem.

|  |  |  |
| --- | --- | --- |
|  |  | (3.1) |

Where maps the features into a higher dimensional space and C is the regularization parameter and is a value greater than 0. Because of the high dimensionality of the value w, then we need the solution of the dual problem.

|  |  |  |
| --- | --- | --- |
|  |  | (3. ) |

Where is the vector of all ones, is an positive semi definite matrix.

, and represents the kernel function.

The problem in 3.2 above is solved using the primal- dual relationship, the required required satisfies

|  |  |  |
| --- | --- | --- |
|  |  | (3. ) |

Now the decision function is given by

|  |  |  |
| --- | --- | --- |
|  |  | (3. ) |

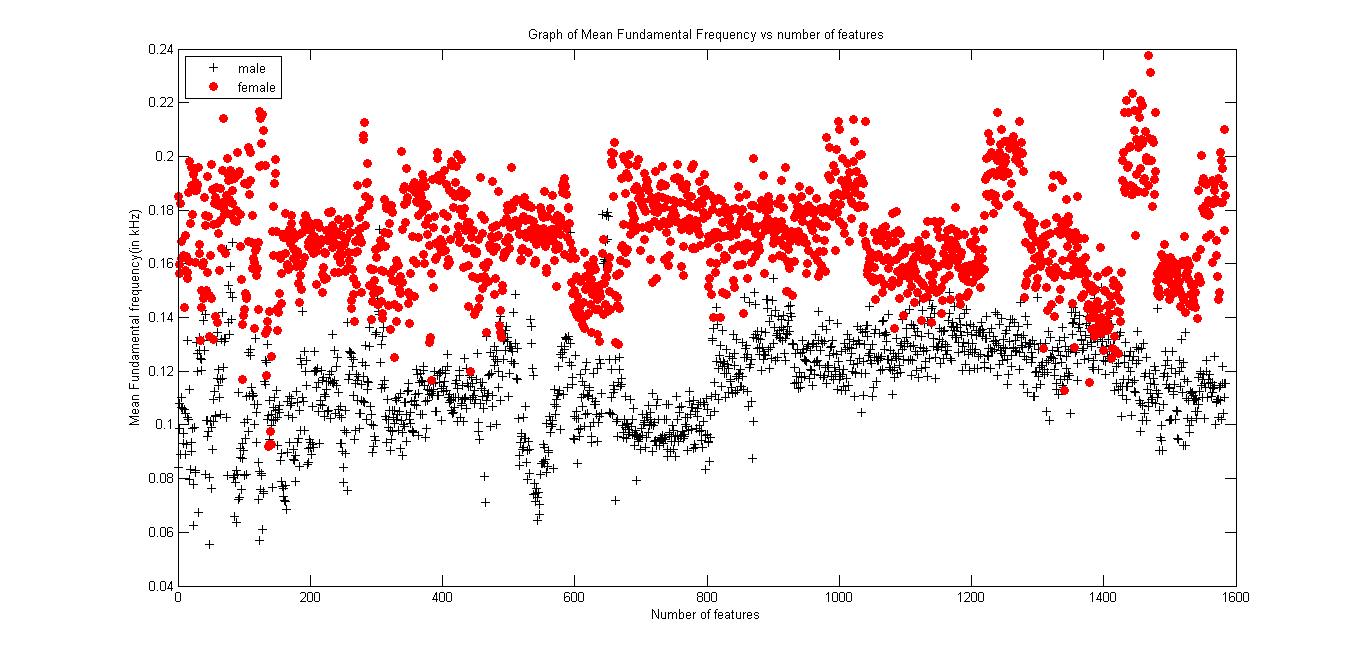
The parameters label names, support vectors, and other selected information is stored in the model for prediction.

The system outputs are

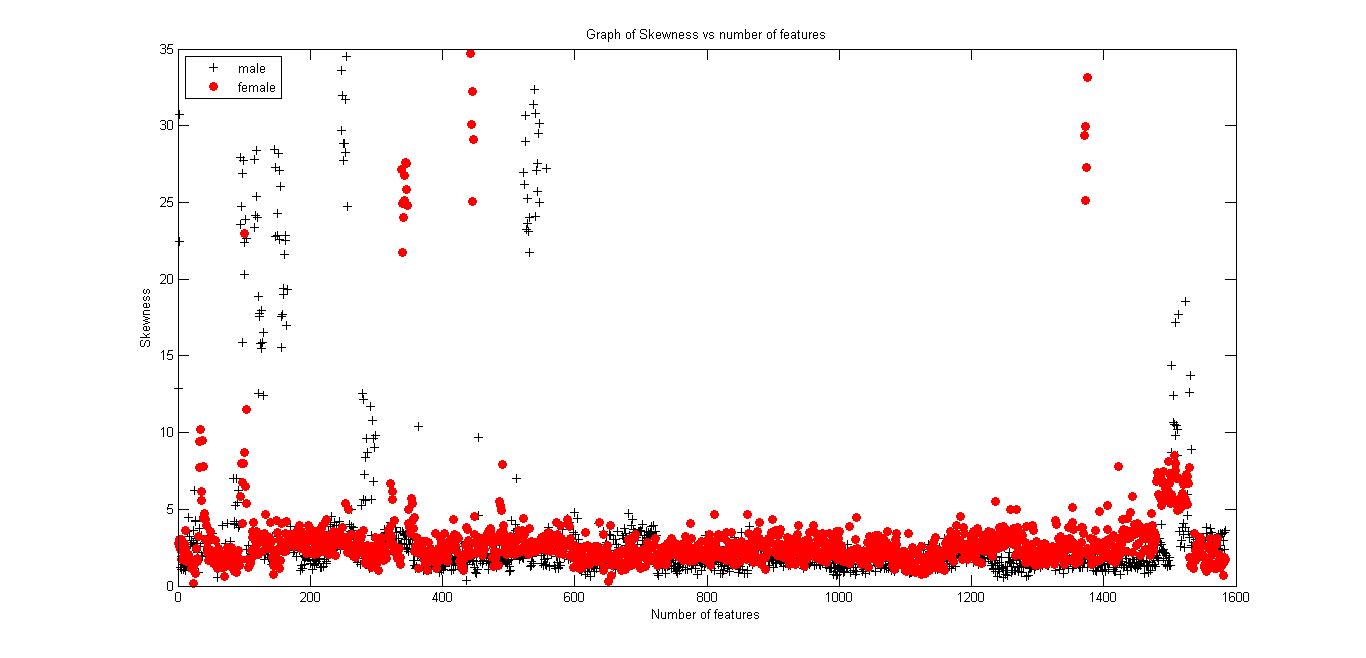
* #iter – number of iteration
* nu – for c-SVM it is equivalent to c, the regularization parameter
* obj – is the optimal value of the dual SVM problem
* rho – is the bias term in the decision function
* nSV-number of support vectors
* nBSV – bounded support vector where alpha
* Total nSV- total number of support vectors

Before feeding into the SVM classifier, the various features were scaled using the function to obtain data that has the same distance metric. The SVM tries to maximize the distance between the hyper plane and support vectors and for feature with high values, they would dominate over others. After scaling the features are fed to the train as well as predict function.

## 3.4 Acoustic Features Model

A dataset of already extracted features from kaggle.com [9] is used. The acoustic properties are mainly statistical in nature, derived from computed weighted mean frequency of speech signals. It consists of 3168 training examples obtained using the *specan* command in R. Plots of feature value versus number of features were made for both male and female for visualization purposes. This enabled main features selection. The main features were found to be; Mean Fundamental Frequency, Interquartile Range, Spectral Entropy, First quartile and Standard deviation. 

##### Fig 3.2: Plot of mean fundamental frequency vs number of features (main feature)



##### Fig 3.3: Plot of skew vs number of samples (not main feature)

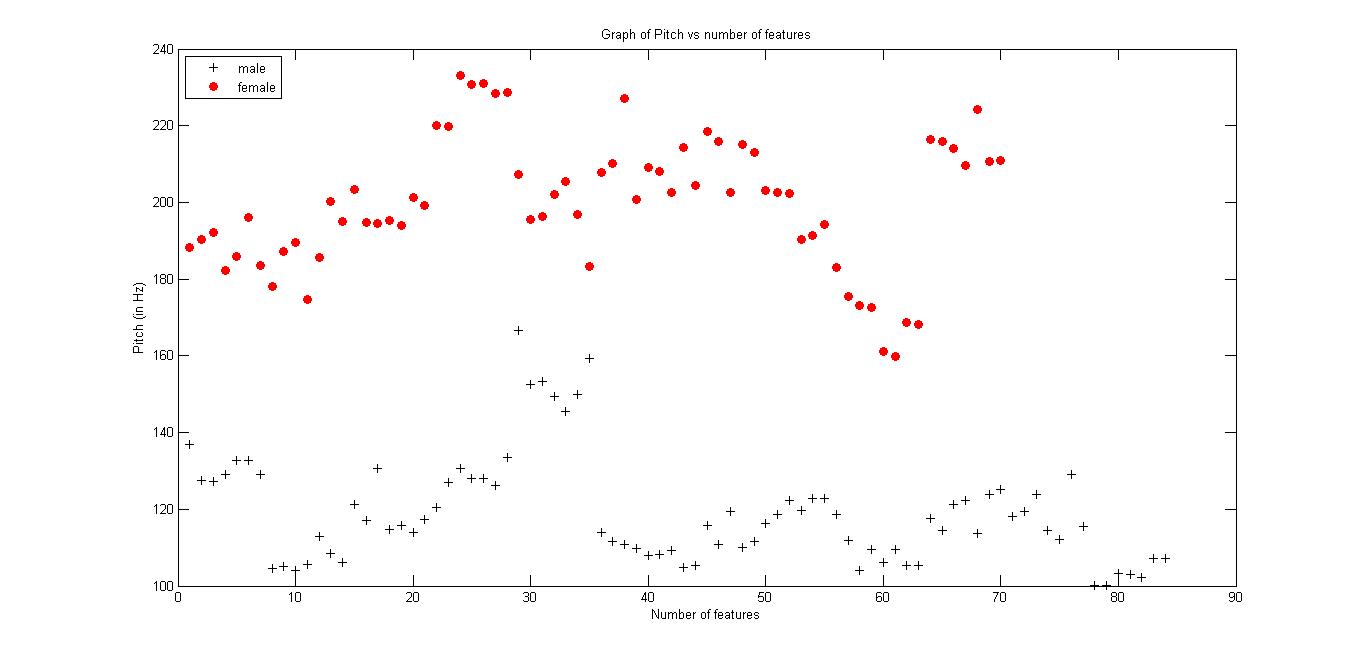
Below are two graphs to show the difference between a main feature, and a ‘minor’ feature. The features were divided into training and test data and fed to an SVM for classification one feature at a time and then a combination of selected features.

The SVM is a powerful classification tool as it combines the accuracies of one feature and another to obtain higher accuracies. Analysis of these combination accuracies is done in details in the next chapter.

## 3.5 Pitch Based Model

The first approach in the design stage was to develop a pitch based model. The speech dataset used was obtained from ELSDSR database [14]. The database consists of 22speakers, 12 male and 10 female. Each of the speaker 7 recording in the train folder and 2 recordings in the test folder. The wav files from train and test folder were each loaded using the function extract \_features. For speech enhancement, the specsub function from matlab voice box toolbox was applied to the input speech files. The pitch of each of the various speakers was obtained using the function pitch\_main. After extraction of pitch, the data was labeled with the labels 1 for male and 0 for female.

A plot of pitch values versus number of examples of each gender were plotted on one plot as below.



##### Fig 3.4: Plot of pitch vs number of samples (ELSDSR train dataset)

From the graph, it can be seen that a threshold of 160 is a good choice of a decision linear boundary. This means making a decision using a linear algorithm would perform well on our training set. The SVM linear kernel was applied and accuracies of the order of 98% were obtained. The recognition rate is found to be quite high on test data but when tested on datasets outside of the ELSDSR, the accuracies dropped significantly.

## 3.6 MFCC Based Model

After loading the wav files on matlab and performing spectral subtraction, MFCCs were extracted using the matlab voice box toolbox function melcepst. The speech files are divided into speech frames, and then MFCC extraction is done on each of the frames. The coefficients are computed from a cosine transform of the logarithm of the short- term power spectrum on the Mel Frequency scale.

The output for each wav file is a matrix where n is the number of frames for each speech file. The mean of the coefficients is calculated to obtain a matrix that will be fed to the machine learning classifier.

The final value, a 1x12 of the MFCC is now fed to the SVM classifier. Training accuracies of 100% were obtained. Various tests were performed on the trained model on both the test data from ELSDSR and other raw speech files obtained from my classmates. The performance of the model is evaluated and the results recorded on chapter 4.

# CHAPTER FOUR: RESULTS

## 4.1 Acoustic Features Based Model

Tests were performed using the first model based on acoustic properties on the dataset obtained from kaggle [9]. Training was done on the whole dataset and the training data evaluated using the RBF kernel. The parameters c and sigma were evaluated for optimal performance and set to 1 and 0 respectively. The results for the train performance of all the features are tabulated in the table below.

###### Table 1: Train accuracy of the 20 acoustic Features for c=1, g=0.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Feature | Training accuracy (%) |  | Feature | Training accuracy (%) |
| 1 | Meanfun | **95.39** | 11 | Meandom | 58.99 |
| 2 | IQR | 90.87 | 12 | Maxdom | 61.23 |
| 3 | Q25 | 86.80 | 13 | Mindom | 58.83 |
| 4 | sfm | 72.60 | 14 | Modindx | 52.36 |
| 5 | Sd | 81.5 | 15 | Dfrange | 60.70 |
| 6 | Spent | 75.94 | 16 | Kurt | 52.11 |
| 7 | Centroid | 64.84 | 17 | Skew | 64.88 |
| 8 | Meanfreq | 64.83 | 18 | Maxdom | 61.23 |
| 9 | median | 62.65 | 19 | Q75 | 54.38 |
| 10 | Minfun | 54.13 | 20 | Mode | 68.68 |

The feature mean fundamental frequency (meanfun) gave the highest accuracy of 95.39%.

The dataset was then divided into 80% of train examples representing 2533 and 20% test examples which is 635 examples. The number of male training examples is 1266 and for female is 1267. For the test data there are 317 male examples and 318 female examples. The model is trained and testing is done on the test data using the model obtained. The parameters c and sigma were varied for the test data. The results are recorded for all the features in an attempt to identify the best suitable values of regularization parameter c and sigma g.

###### Table 2: Test accuracies for the 20 acoustic Features c=0.1, g=0.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| c = 0.1, g = 0.1, Training set = 2533, Test set =635 | | | | | | | |
|  | Feature | Kernel | |  | Feature | Kernel | |
| Linear | Gaussian | Linear | Gaussian |
| 1 | Meanfun | **93.07** | **92.59** | 11 | Meandom | 56.85 | 54.80 |
| 2 | IQR | 85.82 | 87.09 | 12 | Maxdom | 56.85 | 58.26 |
| 3 | Q25 | 78.26 | 73.22 | 13 | Mindom | 63.46 | 65.19 |
| 4 | sfm | 65.98 | 65.66 | 14 | Modindx | 40.15 | 40.63 |
| 5 | Sd | 70.23 | 71.81 | 15 | Dfrange | 56.53 | 58.11 |
| 6 | Spent | 71.97 | 68.98 | 16 | Kurt | 55.59 | 56.38 |
| 7 | Centroid | 66.61 | 62.67 | 17 | Skew | 43.46 | 54.33 |
| 8 | Meanfreq | 65.19 | 64.72 | 18 | Maxfun | 50.70 | 49.13 |
| 9 | median | 65.19 | 64.72 | 19 | Q75 | 47.87 | 49.29 |
| 10 | Minfun | 57.95 | 56.37 | 20 | mode | 80.94 | 81.89 |

###### Table3: Test accuracies for the 20 acoustic Features c=1, g=0.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test accuracy (%) c= 1, g = 0.1, Training set = 2533, Test set =635 | | | | | | | |
|  | Feature | Kernel | |  | Feature | Kernel | |
| Linear | Gaussian | Linear | Gaussian |
| 1 | Meanfun | **93.34** | **93.07** | 11 | Meandom | 56.85 | 42.36 |
| 2 | IQR | 85.83 | 85.98 | 12 | Maxdom | 56.69 | 58.27 |
| 3 | Q25 | 78.27 | 71.65 | 13 | Mindom | 63.46 | 68.51 |
| 4 | sfm | 65.98 | 64.56 | 14 | Modindx | 40 | 40.63 |
| 5 | Sd | 70.24 | 66.93 | 15 | Dfrange | 56.53 | 58.27 |
| 6 | Spent | 72.28 | 68.98 | 16 | Kurt | 55.59 | 48.66 |
| 7 | Centroid | 66.61 | 59.53 | 17 | Skew | 46.14 | 60.15 |
| 8 | Meanfreq | 66.61 | 59.52 | 18 | Maxfun | 56.69 | 58.27 |
| 9 | median | 65.19 | 64.25 | 19 | Q75 | 48.35 | 49.60 |
| 10 | Minfun | 58.11 | 57.32 | 20 | mode | 80.94 | 82.83 |

Now the features were combined selectively in an attempt to obtain a better accuracy by using multiple input features. The features meanfun and IQR was the first combination since they performed best on the overall training data accuracy. They were tested on both linear and Gaussian kernel for selected parameter values c and g.

###### Table 4: Combination of different selected features.

|  |  |  |
| --- | --- | --- |
| Test accuracies in (%) for c=0.1, g=0.1, Training set = 2533, Test set =635 | | |
|  | Kernel | |
| Feature | Linear | Gaussian |
| Meanfun+IQR | 93.54 | 93.22 |
| Meanfun+IQR+Q25 | 93.85 | 92.1 |
| Meanfun+IQR+Q25+sd | 94.48 | 92.91 |
| Meanfun+IQR+Q25+sd+meanfreq | 94.64 | 92.59 |
| Meanfun+IQR+Q25+sd+meanfreq+sfm | 93.5 | 91.07 |
| Meanfun+sd+IQR | 94.33 | 92.38 |
| Meanfun+sd+IQR+meanfreq | 94.33 | 92.44 |
| Meanfun+sd+IQR+meanfreq+Q25 | 94.33 | 92.28 |
| Meanfun+sd+IQR+meanfreq+Q25+median | 94.33 | 91.81 |
| All 20 features | 94.33 | 90.7 |

The model shows are increase in training accuracy as well as test accuracy. More features were added and it was observed that the training accuracy increases with addition of individually well performing features and reduces on adding some. The highest accuracies were achieved by combining the features meanfun, IQR, Q25, sd and meanfreq. This formed the final model features and was tested for different values of c and g.

###### Table 5: Test on different parameter values of c and g on the main features.

|  |  |  |
| --- | --- | --- |
| Test accuracy (%) of model based on: Meanfun,IQR,sd,Q25 and Meanfreq | | |
|  | Kernel | |
| Parameters | Linear | Gaussian |
| C=10,g=0.1 | 94.48 | 92.59 |
| C=0.1,g=10 | 94.33 | 66.14 |
| C=0.1,g=0.01 | 94.33 | 92.44 |
| C=1,g=1 | 94.64 | 92.12 |
| C=0.1,g=0.1 | 94.64 | 92.59 |
| C=1,g=0.01 | **94.64** | **92.59** |

The best acoustic model from the combination of the five features is obtaining by using c=1 and g=0.01 with accuracies of 92.59% on using the Gaussian kernel.

## 4.2 Pitch Based Model

A pitch detection Algorithm was implemented using autocorrelation. It was used for feature extraction of the wav files for the train and test data of the ELSDSR database. The train set consisting of 154 features is fed to the SVM classifier and tested on different values of the parameters c and g.

###### Table 6: Train accuracies of pitch model using ELSDSR dataset

|  |  |
| --- | --- |
| Training Accuracy(%) for ELSDSR Training set | |
| Parameter values | RBF kernel |
| C=0.1,g=0.1 | 54.54 |
| C=1,g=1 | 100 |
| C=1,g=0.1 | 100 |
| C=0.1,g=0.01 | 99.35 |
| C=10,g=1 | 96.27 |

The test data was included in the testing of the model accuracy. Now the model trained using the 154 train set is tested using the 44 ELSDSR test set.

###### Table 7: Test model based on Pitch.

|  |  |  |
| --- | --- | --- |
| Test Accuracy(%) for ELSDSR Train set=154,Test set=44 | | |
|  | Kernel | |
| Parameter values | Linear | Gaussian |
| C=0.01, g=0.1 | 97.72 | 54.54 |
| C=1, g=1 | 97.72 | 97.72 |
| C=0.1, g=0.1 | 97.72 | 97.72 |
| C=100, g=0.1 | 100 | 97.72 |
| C=0.1, g=0.01 | 97.72 | 97.72 |
| C=1000, g=0.01 | 100 | 97.72 |

High accuracies is achieved for both linear and RBF kernel with the linear kernel giving better results. The high accuracies maybe because the speech files contained in both the train and test folders are from the same speakers. The test set which has 44 examples is used as the train set and the train set used as the test set.

###### Table 8: Performance of pitch for a smaller training set.

|  |  |  |
| --- | --- | --- |
| Test Accuracy(%) for ELSDSR Train set=44, Test set=154 | | |
|  | Kernel | |
| Parameter values | Linear | Gaussian |
| C=0.01, g=0.1 | 98.05 | 54.54 |
| C=1, g=1 | 98.70 | 81.16 |
| C=0.1, g=0.1 | 98.70 | 54.54 |
| C=100, g=0.1 | 98.70 | 96.10 |
| C=1, g=0.1 | 98.70 | 96.10 |
| C=1000, g=0.01 (over fit) | 98.70 | 98.70 |

The linear model achieves higher accuracies and is less affected by the changes in c and g. high values of c as in the case of c=1000 tend to give accurate results but the model is over fitted. The opposite is true as low values of c under fits the model and thus low accuracies achieved. The parameter for the pitch based model was chosen as c=1 and g=0.1

Now the model was tested on other recordings that are not from the ELSDSR database. The set is made up of noisy voice recordings obtained from recording using matlab audio recorder.

###### Table 9: Performance of pitch when tested on raw recorded files

|  |  |  |
| --- | --- | --- |
| Test Accuracy (%) for raw recordings. Train set=154, Test set=18 | | |
|  | Kernel | |
| Parameter values | Linear | Gaussian |
| C=0.01, g=0.1 | 88.88 | 55.55 |
| C=1, g=1 | 83.33 | 83.33 |
| C=0.1, g=0.1 | 83.33 | 83.33 |
| C=100, g=0.1 | 88.88 | 88.88 |
| C=1, g=0.1 | 83.33 | 83.33 |
| C=1000, g=0.01 (over fit) | 88.88 | 88.88 |

## 4.3 MFCC Based Model

The extracted MFCC features are input to the SVM classifier and the training accuracy obtained for all the training data for different values of c and g.

###### Table 10: Train model based on MFCC.

|  |  |
| --- | --- |
| Training Accuracy(%) for ELSDSR Training set | |
| Parameter values | RBF kernel |
| C=0.1,g=0.1 | 100 |
| C=0.01,g=0.1 | 54.54 |
| C=0.05,g=0.1 | 96.10 |
| C=0.1,g=10 | 54.54 |
| C=0.1,g=0.2 | 98.70 |
| C=0.1,g=0.3 | 55.19 |

Then test data was now tested using the MFCC model.

###### Table 11: Test model based on MFCC.

|  |  |  |
| --- | --- | --- |
| Test Accuracy(%) for ELSDSR Train set=154, Test set=44 | | |
|  | Kernel | |
| Parameter values | Linear | Gaussian |
| C=0.01, g=0.1 | 100 | 54.54 |
| C=1, g=1 | 100 | 79.54 |
| C=0.1, g=0.1 | 100 | 100 |
| C=0.05, g=0.1 | 100 | 93.18 |
| C=0.1, g=0.01 | 100 | 100 |
| C=100, g=1 | 100 | 79.54 |

For this case, the linear kernel achieves 100% on the test data. This is explained by the similarity in test and train data. In real world situations, there would be variations in accuracies for different voice samples introduced into the system. Use of a smaller dataset for training and testing helps in visualization of change in accuracies for different cases.

###### Table 12: Test model based on MFCC (smaller training set).

|  |  |  |
| --- | --- | --- |
| Test Accuracy(%) for ELSDSR Train set=44,Test set=154 | | |
|  | Kernel | |
| Parameter values | Linear | Gaussian |
| C=0.01, g=0.1 | 99.35 | 54.54 |
| C=10, g=0.01 | 98.70 | 98.05 |
| C=0.1, g=0.1 | 98.70 | 54.54 |
| C=0.3, g=0.1 | 99.35 | 86.36 |
| C=1, g=0.1 | 98.70 | 99.35 |
| C=100, g=1 | 98.70 | 59.09 |

For MFCC a robust model is developed with 44 train examples and on the 154 test examples. The linear model gives high accuracies and does not vary much with changes in c and g, however the best accuracy is achieved by using the RBF model.

###### Table 13: Performance of MFCC model when tested on raw recorded files.

|  |  |  |
| --- | --- | --- |
| Test Accuracy (%) for raw recordings. Train set=154, Test set=18 | | |
|  | Kernel | |
| Parameter values | Linear | Gaussian |
| C=0.01, g=0.1 | 83.33 | 55.55 |
| C=1, g=1 | 77.77 | 55.55 |
| C=0.1, g=0.1 | 77.77 | 83.33 |
| C=100, g=0.1 | 77.77 | 77.77 |
| C=1, g=0.1 | 77.77 | 83.33 |
| C=1000, g=0.01 (over fit) | 77.77 | 77.77 |

## 4.4 Combined MFCC and Pitch Model

The features pitch and MFCC were combined.

###### Table 14: Train model based on MFCC and pitch.

|  |  |
| --- | --- |
| Training Accuracy(%) for ELSDSR Training set | |
| Parameter values | Gaussian kernel |
| C=0.1,g=0.1 | 100 |
| C=0.01,g=0.1 | 54.54 |
| C=10,g=0.1 | 100 |
| C=0.1,g=10 | 54.54 |
| C=0.1,g=0.2 | 98.70 |
| C=0.1,g=0.3 | 100 |

Now the performance on test data is investigated.

###### Table 15: Test model based on MFCC and pitch

|  |  |  |
| --- | --- | --- |
| Test Accuracy(%) for ELSDSR Train set=154, Test set=44 | | |
|  | Kernel | |
| Parameter values | Linear | Gaussian |
| C=0.1, g=0.1 | 100 | 100 |
| C=1, g=1 | 100 | 79.54 |
| C=1, g=0.01 | 100 | 100 |
| C=0.1, g=0.01 | 100 | 100 |
| C=0.1, g=0.5 | 100 | 54.54 |
| C=100, g=1 | 100 | 79.54 |

Cases of 100% accuracy are obtained also for the case of combined pitch and MFCC. The same smaller set of training set is used for visualization.

###### Table 16: Test model based on MFCC and pitch (smaller training set)

|  |  |  |
| --- | --- | --- |
| Test Accuracy(%) for ELSDSR Train set=44,Test set=154 | | |
|  | Kernel | |
| Parameter values | Linear | Gaussian |
| C=0.1, g=0.1 | 100 | 54.54 |
| C=10, g=0.01 | 100 | 100 |
| C=1, g=0.1 | 100 | 100 |
| C=0.3, g=0.1 | 100 | 100 |
| C=1, g=0.1 | 100 | 100 |
| C=100, g=1 | 100 | 59.09 |

It is seen that by combining MFCC and pitch we obtain quite high accuracies on doing an equivalent comparison with pitch alone, and MFCC alone. This applies for choice of parameters c between 0.3 and 10, and g between 0.01 and 0.1. The combined MFCC and pitch model forms the final model as it gives the highest accuracy on test data.

###### Table 17: Test model based on MFCC and pitch on raw recorded files

|  |  |  |
| --- | --- | --- |
| Test Accuracy (%) on raw recordings. Train set=154, Test set=18 | | |
|  | Kernel | |
| Parameter values | Linear | Gaussian |
| C=0.01, g=0.1 | 83.33 | 55.55 |
| C=1, g=1 | 83.33 | 55.55 |
| C=0.1, g=0.1 | 83.33 | 88.88 |
| C=100, g=0.1 | 83.33 | 88.88 |
| C=1, g=0.1 | 83.33 | 88.88 |
| C=1000, g=0.01 (over fit) | 83.33 | 88.88 |

When applied on raw recordings however, the accuracy does not improve by a good margin. This is because MFCCs are not very robust in the presence of additive noise.

# CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

## 5.1 CONCLUSION

This project objective was to demonstrate gender detection by speech. The necessary background procedures required to run such a system such as speech signal processing and classification were studied and discussed. As expected, pitch was found to be good discriminator in gender detection but is not an ultimate base for such a decision. By incorporating MFCC as a feature along with pitch it was observed the accuracies increased and thus the final model was based on a combination of both features. When used on noisy speech files the MFCC does not improve the system accuracy they are sensitive to noise.

SVM as a classification algorithm gave satisfactory results. The dataset used to train the model, was quite small and this shows SVMs robustness given such a task. It is observed that use of a greater number of feature example develops a more learned model and thus better performance.

The final implementation is able to identify the gender of a person through the computer microphone.

## 5.2 RECOMMENDATIONS

Some recommendations for further work on the project include;

* Applying other machine learning algorithms for classification such as neural networks which may be more reliable in the long run.
* Adding a graphical user interface that the user doesn’t have to deal with script to perform different system alterations.
* Use of larger datasets to obtain a more robust model
* Using other features of speech like shifted delta cepstral coefficients for classification purposes.

# REFERENCES

|  |  |
| --- | --- |
| [1] | V. k. Madisetti and D. B. Williams, The Digital Processing Handbook, Florida: CRC Press, Inc., 1999. |
| [2] | N. Jain and D. Kaushik, "GENDER VOICE RECOGNITION THROUGH SPEECH ANALYSIS WITH HIGHER ACCURACY". |
| [3] | H. T. a. A. Eriksson, "The frequency range of the voice fundamental in the speech of," 1995. |
| [4] | S. R. Kulkarn, Sampling and Quantization, 2002. |
| [5] | G. Erickson, "A Fundamenta Instroduction to the compact disk player," 1994. |
| [6] | H. H. a. N. Morgan., "Rasta processing of speech.," *Speech and Audio Processing,* vol. IEEE Transactions on, p. 2(4):578–589, 1994. |
| [7] | L. T. a. M. Karnjanadecha, "PITCH DETECTION ALGORITHM: AUTOCORRELATION METHOD AND AMDF," 2003. |
| [8] | K. Prahallad, *Speech Technology.* |
| [9] | "kaggle Inc," [Online]. Available: https://www.kaggle.com/primaryobjects/voicegender. |
| [10] | A. Ng, "Stanford Machine Learning," 2012. |
| [11] | I. G. a. Y. B. a. A. Courville, Deep Learning, 2016. |
| [12] | C. M. Bishop, Pattern Recognition adn Machine Learning, 2006. |
| [13] | a. C. J. L. C. C. Chang, "LIBSVM: a library for support vector machines," 2001. |
| [14] | Feng.L, "Speaker Recognition, Informatics and Mathematical Modelling," *Technical University of Denmark, DTU,* 2004. |
| [15] | M. Brookes, "Voicebox: Speech processing toolbox for matlab," Mar 2011. |

# APPENDIX

### Acoustic model main code

clc; clear all;

%load dataset to the environment

load ('kaggledata.mat');

%define input features and and label

y=label;

%main features

% X=mode;

%test on the five main features

X=[meanfun IQR sd meanfreq Q25];

%test for all the features

% X=[meanfun IQR sfm centroid Q25 sd meanfreq Q75 median spent maxdom ...

% mindom minfun maxfun mode meandom modindx meandom meanfun skew];

% scaling of the features

[Xscaled] = feature\_scale(X);

x=Xscaled;

%========================================================================

%divide data into training and testing set

train\_data = x(318:2850,:);

train\_label = y(318:2850,:);

test\_data\_m= x(1:317,:); test\_label\_m =y(1:317,:); %male

test\_data\_f=x(2851:3168,:); test\_label\_f=y(2851:3168);%female

test\_data=[test\_data\_m;test\_data\_f];

test\_label=[test\_label\_m;test\_label\_f];

%=========================================================================

%training with 80% and testing on 20% of the dataset(linear)

disp('==============Test accuracy using Linear Kernel=================');

%the options -t 0 represents choice of linear kernel

model\_Linear = svmtrain(train\_label, train\_data, '-t 0 -c 0.1 -g 0.1 -h 0');

[predict\_label\_L, accuracy\_L, dec\_values\_L] = svmpredict(test\_label,...

test\_data, model\_Linear);

%======================================================================

%training with 80% and testing on 20% of the dataset(RBF)

disp('==============Test accuracy using GAUSSIAN Kernel================');

%the options -t 2 represents choice of gaussian kernel

model\_RBF\_test = svmtrain(train\_label,train\_data, '-t 2 -c 0.1 -g 0.1 -h 0');

[predict\_label\_R, accuracy\_R, dec\_values\_R] = svmpredict(test\_label,...

test\_data, model\_RBF\_test);

### Pitch, MFCC, combined pitch and MFCC main code

clear all; clc; close all;

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PITCH BASED MODEL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%load dataset for training and testing

load('elsdsr\_train.mat');

%define inputs and labels

X=pitch\_matrix;

%feature scaling

[Xscaled] = feature\_scale(X);

x=Xscaled;

% training data

train\_data = x; train\_label = y;

% =====================================================================

%load test data

load('samples\_test.mat');

X=pitch\_matrix\_s;

%feature scaling

[Xscaled] = feature\_scale(X);

x=Xscaled;

%test data

test\_data=x; test\_label=y\_s;

%======================================================================

%Linear Kernel on train

disp('=======================PITCH BASED MODEL========================');

disp('==============Test accuracy using Linear Kernel=================');

model\_linear = svmtrain(train\_label, train\_data, '-t 0 -c 0.01 -g 0.1 -h 0');

[predict\_label\_L, accuracy\_L, dec\_values\_L] = svmpredict(test\_label,...

test\_data, model\_linear);

% =======================================================================

%Gaussian Kernel

disp('==============Test accuracy using GAUSSIAN Kernel================');

model\_RBF = svmtrain(train\_label, train\_data, '-t 2 -c 0.01 -g 0.1 -h 0');

[predict\_label\_G, accuracy\_G, dec\_values\_G] = svmpredict(test\_label,...

test\_data, model\_RBF);

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MFCC\_MODEL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%load dataset for training and testing

load('elsdsr\_train.mat');

%define inputs and labels

X=mfcc\_matrix;

%feature scaling

[Xscaled] = feature\_scale(X);

x=Xscaled;

% training data

train\_data = x;

train\_label = y;

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*load test data\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

load('samples\_test.mat');

X=mfcc\_matrix\_s;

%feature scaling

[Xscaled] = feature\_scale(X);

x=Xscaled;

%test data

test\_data=x;

test\_label=y\_s;

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Applying Linear kernel on test Data\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

disp('=================================================================');

disp('=======================MFCC BASED MODEL==========================');

disp('===============Test accuracy using Linear Kernel=================');

model\_linear = svmtrain(train\_label, train\_data, '-t 0 -c 0.01 -g 0.1 -h 0');

[predict\_label\_L, accuracy\_L, dec\_values\_L] = svmpredict(test\_label,...

test\_data, model\_linear);

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Applying Gaussian Kernel on Test Data\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

disp('==============Test accuracy using GAUSSIAN Kernel================');

model\_RBF = svmtrain(train\_label, train\_data, '-t 2 -c 0.01 -g 0.1 -h 0');

[predict\_label\_G, accuracy\_G, dec\_values\_G] = svmpredict(test\_label,...

test\_data, model\_RBF);

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COMBINED MFCC AND PITCH\_MODEL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%load dataset for training and testing

load('elsdsr\_train.mat');

%define inputs and labels

X=[mfcc\_matrix pitch\_matrix];

%feature scaling

[Xscaled] = feature\_scale(X);

x=Xscaled;

% training data

train\_data = x; train\_label =y;

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*load test data\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

load('samples\_test.mat');

X=[mfcc\_matrix\_s pitch\_matrix\_s];

%feature scaling

[Xscaled] = feature\_scale(X);

x=Xscaled;

%test data

test\_data=x; test\_label=y\_s;

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Applying Linear kernel on test Data\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

disp('=================================================================');

disp('================COMBINED MFCC AND PITCH MODEL====================');

disp('==============Test accuracy using Linear Kernel==================');

model\_linear = svmtrain(train\_label,train\_data, '-t 0 -c 0.01 -g 0.1 -h 0');

[predict\_label\_L, accuracy\_L, dec\_values\_L] = svmpredict(test\_label,...

test\_data, model\_linear);

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Applying Gaussian Kernel on Test Data\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

disp('==============Test accuracy using GAUSSIAN Kernel================');

model\_RBF = svmtrain(train\_label, train\_data, '-t 2 -c 0.01 -g 0.1 -h 0');

[predict\_label\_G, accuracy\_G, dec\_values\_G] = svmpredict(test\_label,...

test\_data, model\_RBF);