ASHESI UNIVERSITY COLLEGE

EMOTION RECOGNITION FROM SPEECH: AN IMPLEMENTATION IN MATLAB

APPLIED PROJECT

B.Sc. Electrical and Electronic Engineering

Maame Akua Afrakoma Wusu-Ansah

2019

ASHESI UNIVERSITY COLLEGE

EMOTION RECOGNITION FROM SPEECH: AN IMPLEMENTATION IN MATLAB

APPLIED

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Electrical & Electronic Engineering.

Maame Akua Afrakoma Wusu-Ansah

2019

DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

……

Date:

……

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University College.

Supervisor's Signature:

……

Supervisor's Name:

……

Date:

……

Acknowledgements

First and foremost, I would like to thank God for bringing me this far. When I started this project, I did not know what I was doing and if I would be able to do it. I am very grateful to Him for His grace and mercies, for keeping me alive and well to see the end of this project.

I would also like to thank my supervisor, Kofi Adu-Labi, for the constant support he gave and believing in me.

Abstract

Human Computer Interaction now focuses more on being able to relate to human emotions. Recognizing human emotions from speech is an area that a lot of research is being done into with the rise of robots and Virtual reality. In this paper, emotion recognition from speech is done in MATLAB. Feature extraction is done based on the pitch and 13 MFCCs of the audio files. Two classification methods are used and compared to determine the one with the highest accuracy for the data set.

Table of Content

List of Abbreviations

AI – Artificial Intelligence

DFT – Discrete Fourier Transform

FAU - Friedrich–Alexander University Erlangen–Nürnberg

FFT – Fast Fourier Transform

HMM – Hidden Markov Models

HVAC – Heating, Ventilation and Air Conditioning

KNN – K-Nearest Neighbor

LDA – Linear Discriminate Analysis

MFCC – Mel-Frequency Cepstral Coefficients

MLP – Multilayer Perceptron

RAVDESS - Ryerson Audio-Visual Database of Emotional Speech and Song

SSI – Social Signal Interpretation

SVM – Support Vector Machines

List of Figures

List of Tables

CHAPTER 1

1.1 Introduction/Background

Emotions have great influence over the way human beings reason, behave, or think. Emotions trigger reactions and “people desire reactions from others according to their emotion” [1]. Emotion can be perceived through one’s facial expressions or speech. Emotion recognition based on speech is an area which is currently being explored to help in the activities of various industries. For call centres, it can provide employees with information concerning the emotions their voice may portray and for computer- enhanced learning. Research has emerged from automatically recognizing purely acted emotions to more natural emotions. [2] Utilizing voice-based emotion recognition in artificial intelligence (AI) products will boost user experience. [3]

The recent trend of home automation is a way of conveniently controlling interconnected devices. Everything from lights and air conditioners to doors can be controlled using a central panel in the house or through wireless devices using an application or even voice commands. It goes from basic switch controls that turn the lights on/off to creating different environments in the house based on requirements or mood, calling an Uber cab for you, controlling the heating, ventilation and air conditioning (HVAC) systems of the house, Audio Visual entertainment systems etc. and many other advanced functions in the technological aspect.

1.2 Problem Definition

To contribute to human-computer interaction by interpreting emotions through voice. Before, human-computer interaction was predominantly based on rational information processing. With the evolution of computing and Artificial Intelligence (AI), “affective computing” leans into ensuring a greater user experience with computers.

1.3 Objectives of the Project Work

The project aims at the following:

- Speech signal processing (audio segmentation, feature extraction, feature classification)
- Acquiring a large data set to train the model
- Training two different classifiers and testing the classifier

1.4 Expected Outcome

The expected outcome of this project is to be able to successfully classify speech samples into 4 different emotions – happy, sad, angry, neutral.

1.5 Justification/ Motivation of the Project Topic

A major motivation comes from the desire to improve the naturalness and efficiency of human-machine interaction. [4] The introduction of affective computing paves a great way for innovation in user experiences for gaming and learning software. Knowledge about the emotional state can help to connect angry callers of an automatic dialogue system to a human operator, to motivate a student at the right time, or to develop a fun game that is influenced by emotional expressions. [2]. The future of technology is geared towards affective computing with the evolution of robots, AI and mobile phones. With the development of this project, I hope to contribute to the improvement of emotion recognition via speech.

1.6 Scope of Work

The project scope covers emotion recognition based on acoustic properties of speech. Data sets will be recorded or retrieved from sound clips and then classified into the various

emotions. Two different methods of classification will be used to categorize the data. The human emotions to be covered are happiness, anger, disgust, sadness and fear.

1.7 Proposed Chapter Outline

The proposed chapter outline for this research is as below:

Chapter 2: Literature Review

Chapter 3: Design

Chapter 4: Implementation

Chapter 5: Results

Chapter 6: Conclusion

CHAPTER 2: Literature Review

To understand the basis of human emotions, facial expressions and speech are ways of judging a person's emotional state. There are different kinds of emotions which can be expressed by the way a person talks. Speech is a complex signal which contains information about the message, speaker, language and emotions. [4]. Emotional speech recognition is a system which basically identifies the emotional state of human beings from their voice. [4]

Extensive research is still underway to develop an efficient human emotion recognition system. Recently, gesture of emotions in speech to communicate with the machines is an upcoming challenge. The emotions can be observed through the variations in several prosodic (elements of speech that are not individual phonetics segments but are properties of syllables and larger units of speech) parameters of a natural language. Acoustic correlates of emotional speech are often listed in terms of features such as utterance intensity, f0 contour, and voice quality as well as timing and speech rate. It is well established that voice quality and emotion go hand in hand. [5]

Algorithms and feature extraction of speech are some of the numerous ways that speech recognition can be done, and they are based on the acoustic-phonetic approach. Algorithms such as template matching come under the pattern recognition approach, while algorithms that depend on knowledge sources, stochastic of speech signals and neural networks are based on the artificial intelligence approach. [5]

The major steps in speech emotion recognition are audio segmentation, feature extraction and classification into emotional states. EmoVoice is a framework which that allows you to build a real time emotion recognition system from acoustic properties of speech. It is built based on the Social Signal Interpretation (SSI) framework which allows feature extraction, building classifiers and online features. EmoVoice consists of two modules,

one to handle audio segmentation, feature extraction, feature selection, and classification and the other for the online tracking of affect in voice while someone is talking. [2] Audio segmentation involves a way of finding the appropriate acoustic segments to use to be able to classify the speech signal under a basic human emotion.

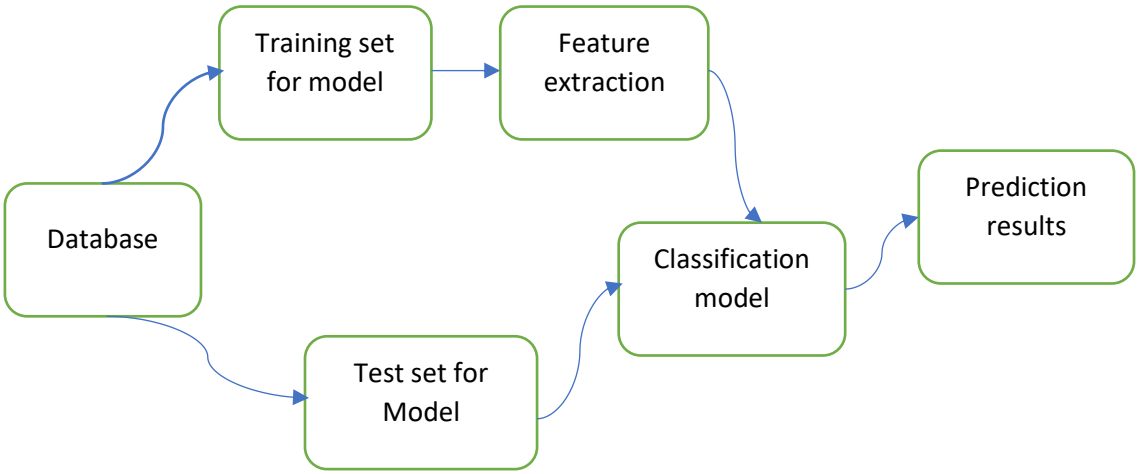

Fig. 2.1. Block Diagram of the proposed Speech Recognition System

2.1 Emotion Units

This sort of classification is done based on utterances, turns or phrases but also on words. [6] Some approaches classify long units, but compute features only from parts of them, for example from the words of an utterance [7, 8, 9] or special phonemes such as vowels or voiced consonants [10, 11]. In order to find suitable phonemes, Busso et al. [12] compared phoneme classes on their relevance with respect to emotions and found vowels to hold more emotional information than for example nasals. Schuller et al. [13] even used phoneme and word specific emotion models. This, however, requires a database with a minimum occurrence of each word. As words may be very short, Batliner et al. [7] based their features on words with a varying number of surrounding words. They also compared turns as classification units with a chunking of the turns by two levels of boundaries. Nicholas et al. [9] classified words, but their annotation was turn-wise, so every word in a turn got the same

emotion label. They also followed a strategy to use only the features of the middle word of a turn to classify the whole turn. In both cases they achieved an improvement over turn-based features. The importance of word position was also shown by the work of Kim et al. [14] who found that sentence medial words are more important than sentence initial and final words.

2.2 Features

The acoustic characteristic of a speech signal is referred to as a feature. Many different speech feature extraction methods have been proposed over the years. Methods are distinguished by the ability to use information about human auditory processing and perception, by the robustness to distortions, and by the length of the observation window. Due to the physiology of the human vocal tract, human speech is highly redundant and has several speaker-dependant features, such as pitch, speaking rate and accent. [15] Features for emotion recognition from speech are calculated from speech as produced by the speaker. Since the speech signal is a waveform and every waveform has properties such as amplitude, time and frequencies, these features can be used to characterise and help distinguish between different emotions. The most commonly used features for speech emotion recognition are derived from prosody, that is energy, pitch and rhythm, hence, features in general are often denoted as prosodic as opposed to linguistic features derived from word information.

The best set of features for automatic classification of emotion has not been established yet. Therefore, on-going research investigates a few different features. Originally, mainly pitch and energy related features were applied, and these continue to be the prominent features.

Table 2.1 An Overview of emotion units and their correlating features. [2]

Unit	Features based on
phonemes	phonemes
words	Words Words in context All phonemes Vowels Voiced consonants
Utterances and turns	Utterances All words Central words

Feature extraction approaches fall in mainly two classes: features computed over short-term speech segments and those computed on global level speech segments. To explain this in more detail, common acoustic feature types such as pitch are usually observations made on short time intervals of about 10–80 ms. These raw values can be used directly for emotion recognition by dynamic classifiers such as Hidden Markov Models (HMMs). Instead, one feature vector for a longer time segment such as a word or an utterance is classified with a static classifier (Support Vector Machines, Neural Networks, etc.). Thus, a series of values must be mapped onto single values to be represented in the feature vector. Timing information is lost by this approach, though it is essential for emotion classification, because emotions are phenomena that do not occur at single points in time only but affect speech continuously and evolve over time. Thus, in static classification, time must be encoded in the feature vector by computing statistical functions such as mean, maximum or minimum from the value series of the basic feature types. [2]

2.2.1 Feature Extraction Using Mel Frequency Cepstrum Coefficient (MFCC)

MFCCs are coefficients that represent audio based on perception with their frequency bands logarithmically positioned and mimics the human vocal response.

The first step in MFCC feature extraction is the frame blocking. Processing of speech signals is done in short time intervals called frames with sizes generally between 20 and 40 ms [16]. Overlapping of frames is done to smoothen the transitions between frames by a predefined size. The first frame consists of the first N = 256 (typical value) samples. The second frame begins at M = 100 (typical value) samples after the first frame, and overlaps it by N−M samples and so on. This process continues until the entire speech signal is accounted. [20] The second step is windowing, where the signal is passed through a windowing function to minimize discontinuities and spectral distortion at the extremes of each frame. The result of windowing a signal x(n) is given by [21],

$$Y(n) = x(n)w(n), \qquad 0 \le n \le N-1 \tag{1}$$

where w(n) is the window function, $0 \le n \le N-1$, and N is the number of samples in each frame. The third step is the Fast Fourier Transform (FFT). FFT is performed on the windowing signal to convert it into frequency domain. The discrete Fourier Transform (DFT) over a discrete signal x(n) of N samples, converts each frame of N samples into the frequency domain from its time domain. The FFT is the fast algorithm to implement DFT, which is defined as

$$X_k = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}}, \qquad k = 0,1,2,3,\ldots,N-1. \tag{2}$$

The result after this step is often referred to as spectrum. The Mel-frequency scale is linearly spaced for frequencies below 1000 Hz and logarithmically spaced above 1000

Hz. The importance of the logarithmic scale appears when using a broad bench of values as it helps to space the small values and approach large values.

The following approximate formula can be used to compute the mels for a given frequency f in Hz [22]:

$$mel(f) = 2595 * \log 10(\ 1 + \frac{f}{700}). \quad (3)$$

2.3 Feature Classification

The aim of a classifier is to designate a class or label from a defined set of classes to an observation. The basic end goal of a classifier is to predict and assign. There are different classifiers for training data that has undergone feature extraction. Among these classifiers are the k-nearest neighbour (K-NN) decision rule, linear discriminate analysis (LDA), Hidden Markov models (HMMs), Neural Networks, MultiLayer Perceptron (MLP), etc. Classifiers (classification algorithms) are chosen based on their ability to handle high-dimensional data, speed and memory usage.

The results in Table 2.3.1 are based on an analysis of many data sets. The data sets in the study have up to 7000 observations, 80 predictors, and 50 classes. This list defines the terms in the table. [23]

Speed: Fast – 0.01 seconds, Medium – 1 second, Slow – 100 seconds

Memory: Small – 1MB, Medium – 4MB, Large – 100MB. [23] Results depend on the type of data and speed of the machine, the table just gives a general guide of the characteristics of the classifiers.

Table 2.3.1 Typical characteristics of the various supervised learning algorithms [24]

Classifier	Categorical Predictor Support	Prediction Speed	Memory Usage
Decision Trees	Yes	Fast	Small
Discriminant analysis	No	Fast	Small for linear, large for quadratic
SVM	Yes	Medium for linear Slow for others	Medium for linear. Medium for multiclass, large for binary
Naïve Bayes	Yes	Medium for simple distributions. Slow for high-dimensional data	Small for simple distributions. Medium for high-dimensional data
Nearest Neighbor	Yes	Slow for cubic Medium for others	Medium
Ensembles	Yes	Fast to medium depending on choice of algorithm	Low to high depending on choice of algorithm

2.3.1 Support Vector Machines

Support Vector Machine is a statistical classifier which classifies data into binary classes (correct or incorrect) based on training data. Support vector machine constructs a hyperplane or set of hyperplanes in a high-dimensional space or infinite dimensional space, which can be

used for classification, regression or other tasks. [4] SVMs are mostly used on acoustic models because they are highly accurate. The idea behind SVM is to find a hyperplane between the instances of two classes in such a way that the shortest distance between the instances and the hyperplane is maximised. Thus, a maximum margin hyperplane is defined on the basis of so-called support vectors, which are training instances situated at the class boundaries. [2]

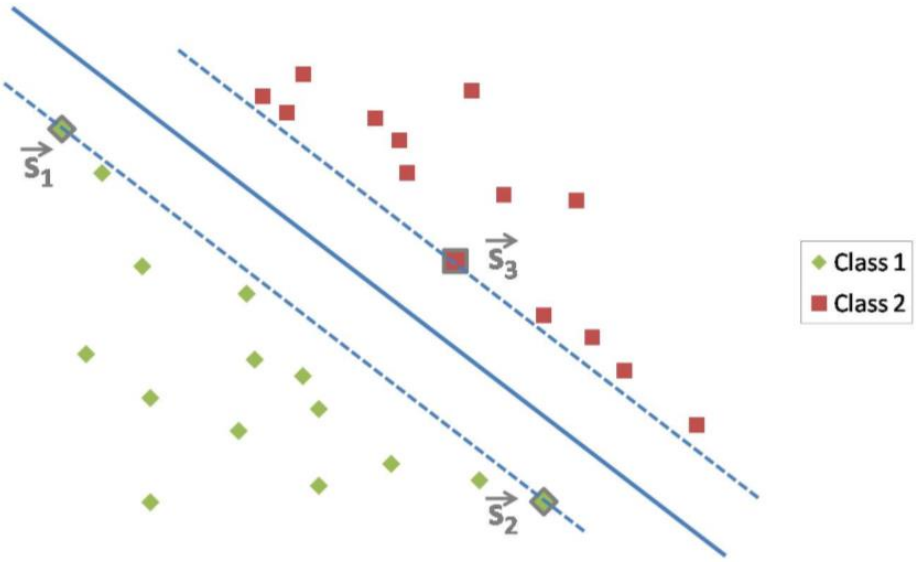

Fig 2.3.1 Support vectors $\overrightarrow{s1}, \overrightarrow{s2}, \overrightarrow{s3}$ for a SVM classifier maximizing the distance between two classes. The solid blue line indicates the maximum margin hyperplane. [2]

2.3.2 K-Nearest Neighbor (K-NN)

KNN is a simple algorithm used in pattern recognition based on memory. The algorithm bases the classification of an unknown sample on the "votes" of K of its nearest neighbor rather than on only it's on single nearest neighbor. When a new sample data x arrives, KNN finds the k neighbors nearest to the unlabelled data from the training space based on some distance [25]. K-NN is a non-parametric classification algorithm. This means that the data is not required to fit a normal distribution. K-NN is used for both regression and classification in pattern recognition. In the classification phase, k is a user-defined

constant, and an unlabelled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point. A commonly used distance metric for continuous variables is Euclidean distance. [26] K-NN as a statistical pattern recognition algorithm does not consider a priori assumption about the distributions from which the training examples are drawn. [25] This means that K-NN does not assume that the data is correct without the need to prove it and so, it involves a training set of all cases. A new sample is classified by calculating the distance to the nearest training case, the sign of that point then determines the classification of the sample. The K-NN classifier extends this idea by taking the K nearest points and assigning the sign of the majority. [25]

CHAPTER 3: Design

3.1 Review of Existing Designs

3.1.1 Databases

Three Databases were used to train the model for EmoVoice – the Berlin database of emotional speech, the SmartKom database and the Friedrich–Alexander University Erlangen–Nürnberg (FAU) Aibo Emotion Corpus. The databases comprise of acted read emotions, voices of both adults and children. All these databases are in German, but the approach works independent of language.

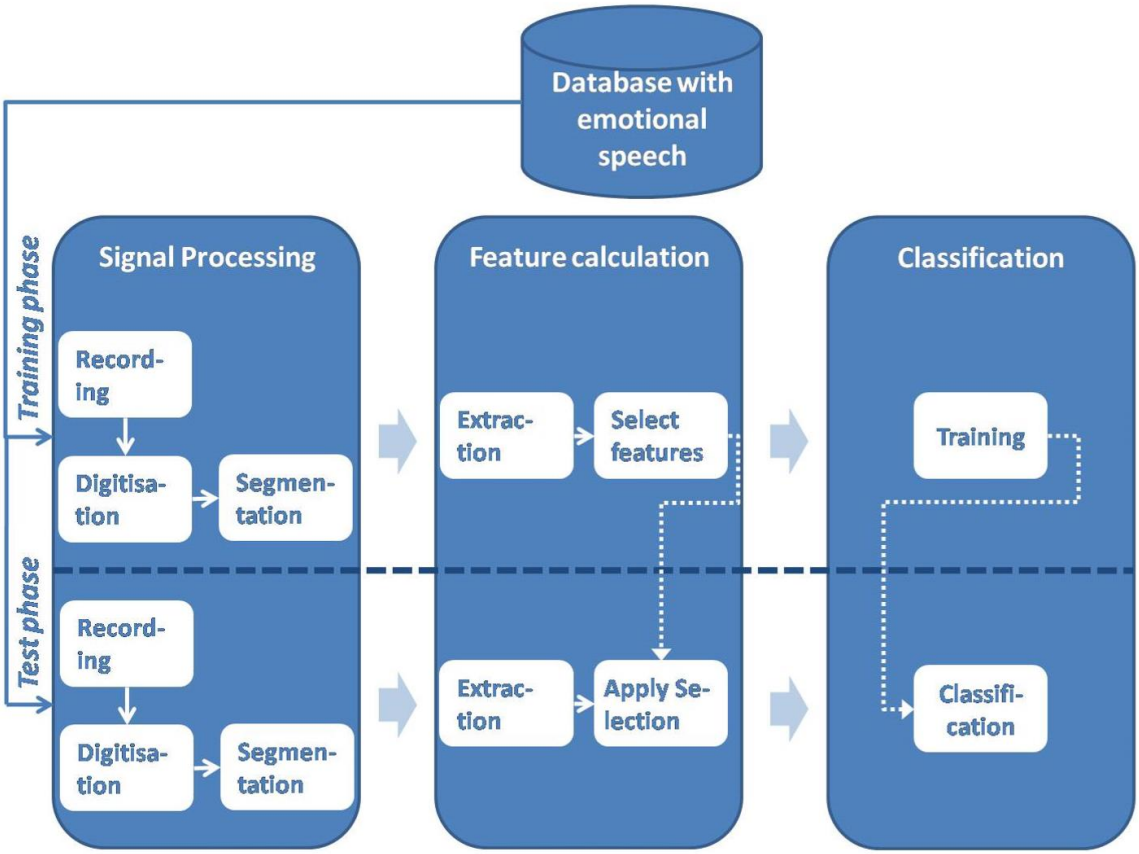

Fig. 3.1.1 Steps in training and testing of the speech emotion recognition system. [4]

The Berlin Database of Emotional Speech contains acted emotional German speech of 5 males and 5 females that were to fake six different emotions – anger, joy, sadness, fear, disgust and boredom as well as a neutral state. [2] The recordings were intended for

phonetic analysis of emotions and emotional speech synthesis and so were conducted under ideal conditions. The FAU Aibo emotion corpus contains speech from children interacting with a remote-controlled robot dog (Aibo). The emotions obtained from the children were spontaneous and short-term. The final categories of emotions were angry, motherese (baby talk), emphatic and neutral. Emphatic and motherese were chosen as emotions because situations with emphatic speech often precede angry situations, and have thus some kind of signalling function, and motherese marks situations when the child wants to compliment Aibo or make it obey in a positive way.

The SmartKom database contains recorded speech from a number of people who were subject to offer input on a mobile communication assistant for the SmartKom project. Even though the speakers did not know that their emotional state was being observed, majority of the speech recorded was emotionally neutral despite the team's efforts to stir up emotions in the speakers. The final categorizations of the emotions were positive, neutral and negative. Because the Berlin database contains acted speech and was recorded under controlled conditions, it is not very practical. Acted emotions can only go so far as to imitate a possibility of an emotion in a certain situation. In addition, it is limited in size. The database records a 100 sound files for each emotion. Both SmartKom and Aibo databases contain larger samples of emotional speech but the emotions in them are not typical or original.

3.1.2 Audio Segmentation

Speech segmentation is a very significant issue because the audio input signal must be put into meaningful units to later derive the actual features from acoustic measurements of those units. Though the decision on which kind of unit to take is evidently important, it has not received much attention in past research on emotion recognition. [2] Most approaches so far have dealt with utterances of acted emotions where the choice of unit is obviously just this

utterance, a well-defined linguistic unit with no change of emotion within in this case. However, in spontaneous speech this kind of obvious unit does not exist. Neither is the segmentation into utterances straight-forward nor can a constant emotion be expected over an utterance. [2] Generally, the decision on which unit to be used as audio segmentation strongly depends on the data. Thus, different types of units are investigated considering their usefulness for different kinds of data. These units are - fixed length units, words, utterances, segments marked by pauses, and turns. An overview of the units for each of the three databases can be found in Table 3.1

Table 3. 1 Emotion units explored for the Berlin, Aibo and SmartKom databases

Unit	**Berlin**	**Aibo**	**SmartKom**
fixed length 0.5s	√	√	√
fixed length 1s	√	√	√
fixed length 2s	√	√	√
automatic pause segmentation by VAD	√	√	√
word	√	√	√
word in context (± 1 word)	√	√	√
manual syntactic/prosodic boundary detection (chunks)	—	√	—
pause segmentation by ASR (chunks)	—	—	√
utterance	√	—	—
turns	—	√	√

Three durations of fixed length units are tested: 0.5, 1 and 2 seconds. These were chosen because units of less than 0.5 seconds were considered as too short for the calculation of statistical measures, while changes of the emotional state may well occur in units longer than 2 seconds. [2] Lastly, as an approximation of a linguistic unit, speech parts segmented by breaks in the voice activity detected automatically and on the acoustic signal only by the algorithm integrated into ESMERALDA, a framework for building automatic speech recognisers based on HMMs [17], are investigated. Words are often very short, that is why they are also investigated within the context of one preceding and succeeding word and the

potential silent or non-verbal part in between. Among the units compared here, considering context is most reasonable for words, because the difference of adjacent words within an utterance in respect of their emotional tone will scarcely be huge. [2] Nevertheless, words are not desirable as a segmentation unit because a speech recognition system is needed to determine word boundaries automatically.

3.2 Thesis Design Objective

The goal is firstly, to find acoustic emotion units that are suitable for real-time applications, secondly, to identify possible acoustic features for emotion recognition that can be extracted fast and automatically, as well as to assess a good procedure to select the most relevant features for a given purpose, and lastly, to choose a fast, but accurate classification algorithm.

CHAPTER 4: Implementation

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) was used as the database for this project.

The RAVDESS is a validated multimodal database of emotional speech and song. The database is gender balanced consisting of 24 professional actors, vocalizing lexically-matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions. Each expression is produced at two levels of emotional intensity, with an additional neutral expression. All conditions are available in face-and-voice, face-only, and voice-only formats. The set of 7356 recordings were each rated 10 times on emotional validity, intensity, and genuineness.

The RAVDESS database gives an index of the emotional corpus through the naming of the files. Files are named with number pairs separated by a hyphen indicating a specific Identifier. For the files used in this project, they were of the form “03-01-04-01-01-01-01.wav”. Where the first two digits represent the modality of the file. ‘01’ represents an audio-visual file, ‘02’ represents a video file, and ‘03’ represents an audio only file. The second pair of digits represents the channel. ‘01’ represents a speech file and ‘02’ represents a song file. The third pair is the emotion identifier. ‘04’ represents Sad, and ‘05’ represents Angry. The remaining pairs represent intensity, the statement used, the repetition and the actor, respectively.

The first proposed approach to emotional recognition from speech was to use “EmoVoice”. EmoVoice is a set of tools that allows you to build your own real-time emotion recognizers from acoustic properties of speech (not using word information). It is implemented with the Social Signal Interpretation (SSI) framework and offers feature extraction, classifier building

and testing, as well as, online recognition. The Social Signal Interpretation (SSI) framework offers tools to record, analyse and recognize human behaviour in real-time, such as gestures, mimics, head nods, and emotional speech. [18]

As this proposed approach brought many challenges, the second proposed approach was an implementation in MATLAB.

Feature Extraction

In MATLAB, Pitch and Mel-Frequency Cepstrum Coefficients (MFCC) are extracted from speech signals recorded for 24 speakers. Speech can be broadly categorized as voiced and unvoiced. In the case of voiced speech, air from the lungs is modulated by vocal cords and results in a quasi-periodic excitation. The resulting sound is dominated by a relatively low-frequency oscillation, referred to as pitch. [19]

Pitch is the quality or degree of highness or lowness of a tone. The first 13 MFCCs were used for evaluating the performance of feature vector. In one scenario, the number of features are reduced to make the model feasible for real-time implementation. As cepstral features are computed by taking the Fourier transform of the warped logarithmic spectrum, they contain information about the rate changes in the different spectrum bands. Cepstral features are favorable due to their ability to separate the impact of source and filter in a speech signal. In other words, in the cepstral domain, the influence of the vocal cords (source) and the vocal tract (filter) in a signal can be separated since the low-frequency excitation and the formant filtering of the vocal tract are located in different regions in the cepstral domain. If a cepstral coefficient has a positive value, it represents a sonorant (a sound produced with the vocal cords so positioned that spontaneous voicing is possible) sound since the majority of the spectral energy in sonorant sounds are concentrated in the low-frequency regions. On the other hand, if a cepstral coefficient has a negative value, it represents a fricative sound since

most of the spectral energies in fricative sounds are concentrated at high frequencies. The lower order coefficients contain most of the information about the overall spectral shape of the source-filter transfer function. The zero-order coefficient indicates the average power of the input signal. The first-order coefficient represents the distribution spectral energy between low and high frequencies. Even though higher order coefficients represent increasing levels of spectral details, depending on the sampling rate and estimation method, 12 to 20 cepstral coefficients are typically optimal for speech analysis. Selecting a large number of cepstral coefficients results in more complexity in the models. For example, if we intend to model a speech signal by a Gaussian mixture model (GMM), if a large number of cepstral coefficients is used, we typically need more data in order to accurately estimate the parameters of the GMM.

A total number of 320 files were processed with 80 files in each emotion category.

In MATLAB the "HelperComputePitchAndMFCC" function was used to extract Pitch and MFCC features. First, the audio samples are collected into frames of 30ms with an overlap of 75%. For each frame, a function is used to decide whether the samples correspond to a voiced speech segment. The pitch and 13 MFCCS (with the first MFCC coefficient replaced by log-energy of the audio signal) are computed for all the audio files. Then the pitch and MFCC information pertaining to the voiced frames only are kept. [19].

Classification

After collecting features for both emotions, a classifier can be trained based on them. Matlab contains a classification learner app to help with the training of data to build a classifier.

For the type of data and features extracted, MATLAB suggests all SVM classifiers – Linear SVM, Cubic, Quadratic, Fine Gaussian, Medium Gaussian and Coarse Gaussian. It also suggests decision trees for the type of data and features extracted. The two classification

methods used in this project are Linear SVM and K-NN. They were both trained using the training data set. After training of the data set was completed, a model was exported and used for the testing data set.

CHAPTER 5: Results

5.1 Results from feature extraction

The data set was divided into training and testing sets. Training Set: 70% and Testing Set: 30%. Fourteen features were extracted from each audio file – the pitch and thirteen (13) MFCCs.

Filename	Pitch	MFCC1	MFCC2	MFCC3	MFCC4	MFCC5	MFCC6	MFCC7
'03-01-04-01-02-01-08.wav'	72.727	-5.1541	2.6965	1.3328	1.3111	0.27229	1.1433	0.15113
'03-01-04-01-02-01-08.wav'	70.796	-5.3222	2.7771	1.7789	1.5666	0.16469	1.4876	0.15141
'03-01-04-01-02-01-08.wav'	89.385	-4.9158	3.8847	2.0763	1.2211	-0.064835	1.1565	-0.18717
'03-01-04-01-02-01-08.wav'	92.486	-4.8583	4.7731	2.3444	0.99745	-0.43548	1.7535	-0.32293
'03-01-04-01-02-01-08.wav'	195.12	-4.9578	4.4937	2.2357	1.1053	-0.1875	1.6696	-0.30391
'03-01-04-01-02-01-08.wav'	197.53	-4.9001	3.6727	2.1672	1.589	0.14705	1.3254	0.048843
'03-01-04-01-02-01-08.wav'	68.966	-5.329	3.4071	2.1211	1.6511	-0.04413	1.216	-0.2513
'03-01-04-01-02-01-08.wav'	70.796	-5.3173	3.8998	2.3199	1.6013	0.39289	1.57	-0.28125

Fig. 5.1 Results obtained from feature extraction

It is observed that the pitch and MFCCs are not on the same scale. To prevent it from biasing the classifier, the features are normalized by subtracting the mean and dividing the standard deviation of each column.

Filename	Pitch	MFCC1	MFCC2	MFCC3	MFCC4	MFCC5	MFCC6	MFCC7
'03-01-04-01-02-01-08.wav'	-0.60938	-1.4141	-1.1118	0.8647	1.2549	0.78335	0.74361	0.010872
'03-01-04-01-02-01-08.wav'	-0.62831	-1.52	-1.033	1.4116	1.6475	0.60476	1.3465	0.01143
'03-01-04-01-02-01-08.wav'	-0.44603	-1.2641	0.049075	1.7761	1.1166	0.2238	0.76656	-0.67172
'03-01-04-01-02-01-08.wav'	-0.41563	-1.2279	0.91696	2.1047	0.77302	-0.39137	1.8121	-0.94563
'03-01-04-01-02-01-08.wav'	0.59083	-1.2906	0.64405	1.9715	0.93875	0.020216	1.6652	-0.90727
'03-01-04-01-02-01-08.wav'	0.61446	-1.2543	-0.15798	1.8875	1.6819	0.57547	1.0624	-0.19552
'03-01-04-01-02-01-08.wav'	-0.64627	-1.5243	-0.41752	1.831	1.7773	0.25817	0.87077	-0.8011
'03-01-04-01-02-01-08.wav'	-0.62831	-1.5169	0.06384	2.0748	1.7008	0.98351	1.4907	-0.86154

Fig. 5.2 Results from normalizing the extracted features

5.2 Results from Classification

The Linear SVM Classifier reported an overall accuracy of 43%. Prediction speed was recorded at 220000 observations per second. Overall training time was recorded at 1015.6 seconds.

Fig. 5.3 shows a scatter plot of the first MFCC against the Pitch. The crosses (**x**) represent incorrect model predictions and the dots (●) represent correct model predictions. For all the graphs plotted, blue represents Angry, Orange – happy, yellow – neutral and sad - purple.

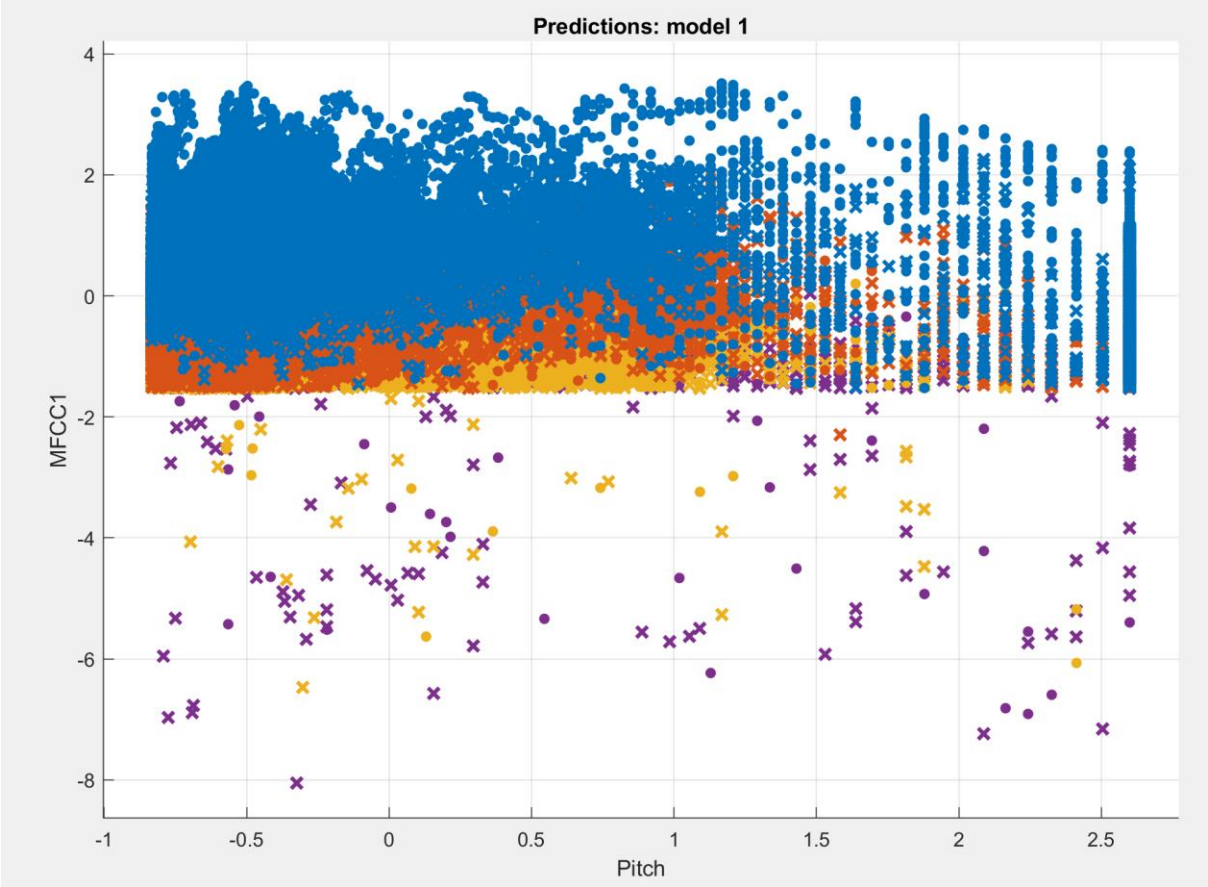

Fig 5.3 Model predictions of Linear SVM Classifier

In Fig. 5.4, the columns show the predicted emotions and the rows show the true classes. In the top row, 59% of the observations made are correctly classified, hence 59% is the true positive rate for correctly classified data points in the Angry class. The other points in the Angry row are misclassified: 22% of the data points were incorrectly classified as Happy, 13% as Neutral and 6% as Sad. 41% is the false negative rate for incorrectly classified data points in the Angry class. In the second row, 43% of the observations made are correctly classified. This is not good enough since it falls below 50%. More data points are classified incorrectly as being other emotions than happy. In the third row, 45% of the data points are

correctly classified whiles 55% are misclassified. This is also not good enough since the misclassifications outweigh the true classifications.

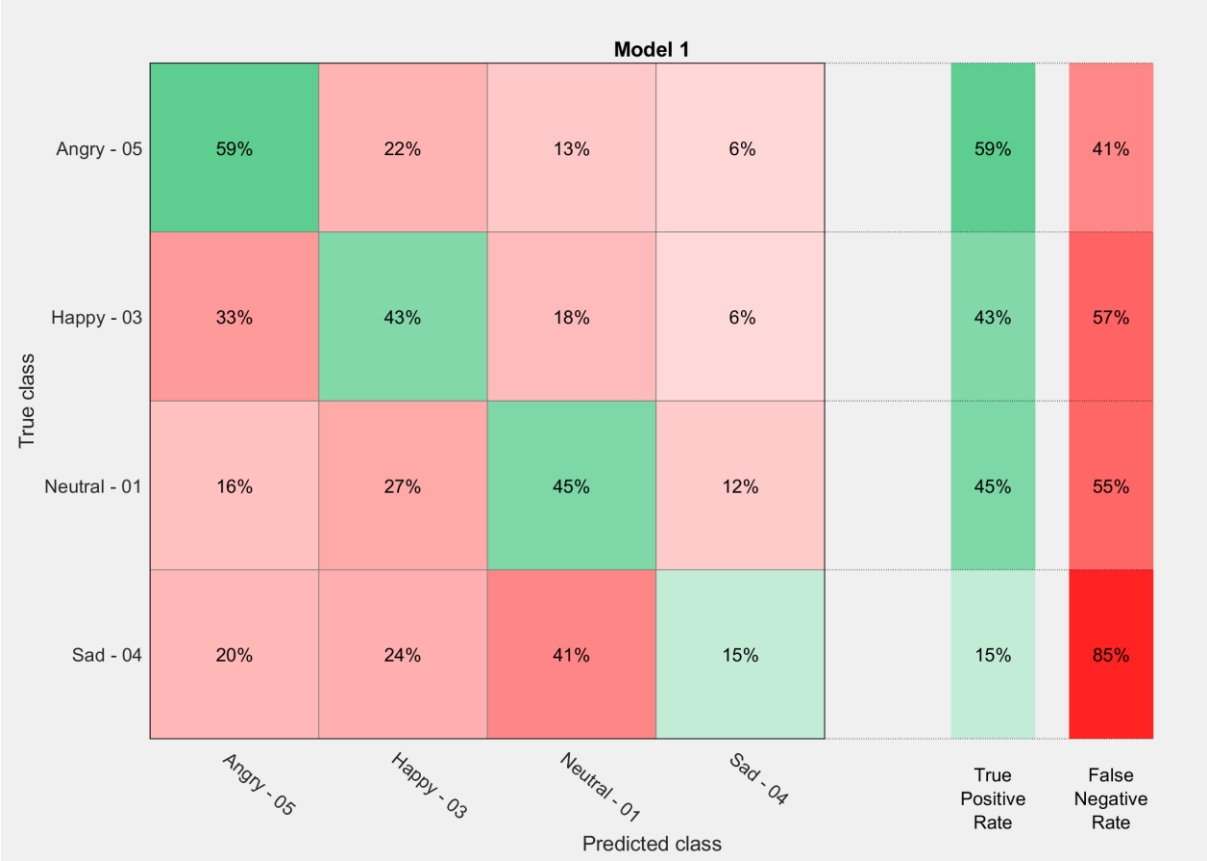

Fig 5.4 Confusion matrix of True Positive rates and False Negative Rates

The Sad class has the worst rate of classifications with a true positive rate of 15%. This means that data points are more likely to be classified as other emotions instead of being correctly classified as Sad.

Overall, the Linear SVM classification method for this data set is lower than average and not that accurate.

Table 5.1 Emotion Recognition Rate with the Linear SVM model

Emotions	Accuracy
Angry	59%
Happy	43%

Neutral	45%
Sad	15%

The KNN classifier reported an overall accuracy of 86.8% . Prediction speed was recorded at 1900 observations per second. Overall training time was recorded at 55.77 seconds.

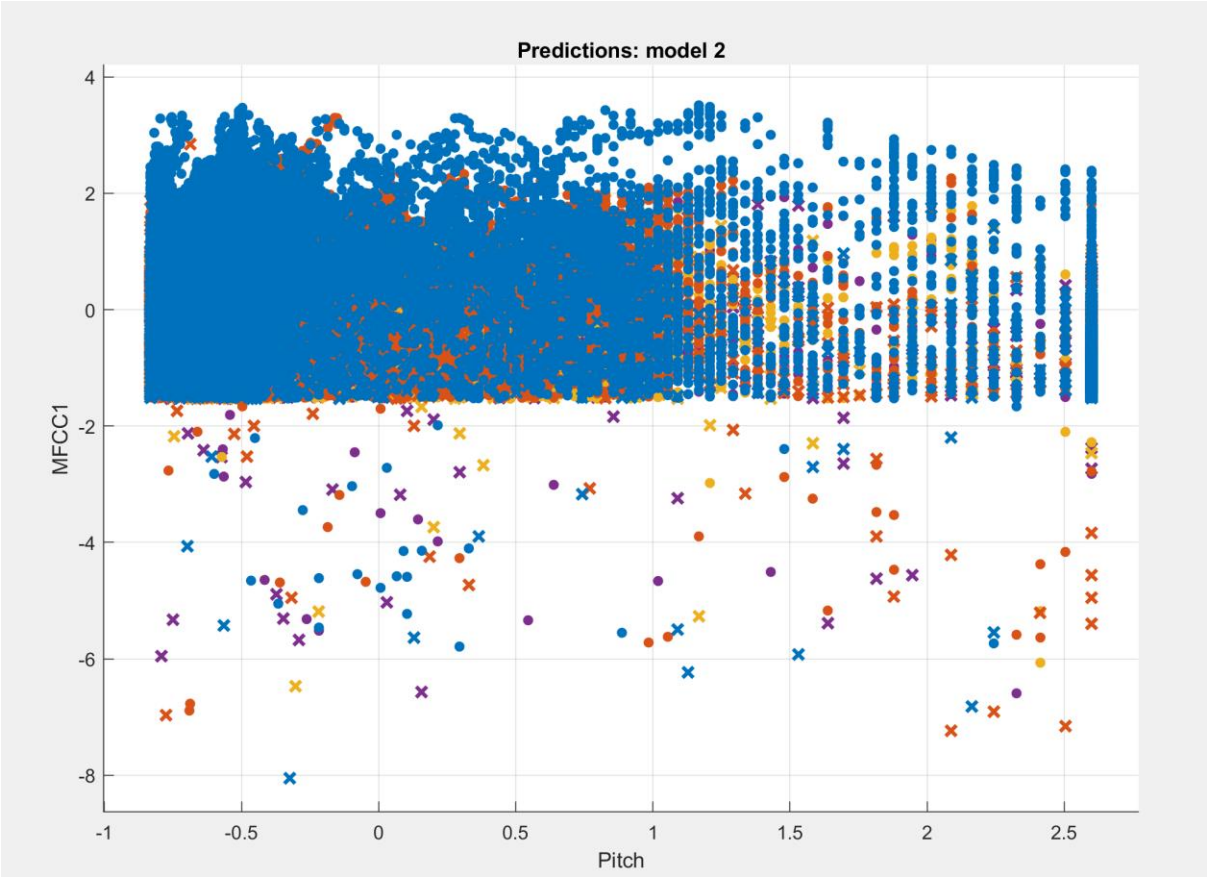

Fig.5.5 Model Predictions of Fine KNN classifier

In Fig.5.5 the scatter plot shows the KNN model's predictions of correctly classified and misclassified data points.

In the top row of fig. 5.6, 89% of the observations made are classified correctly, and so 89% is the true positive rate for correctly predicted points in the Angry class. This model's accuracy is higher than the linear SVM's true positive rate for the Angry class. In the second row, the model presents an 87% true positive rate for correctly classified data points in the Happy class. Only 13% of the data points are misclassified as being other

than Happy. In both the third and last rows, the true positive rate for correctly predicted points are 85%. 15% of the points are misclassified in each case.

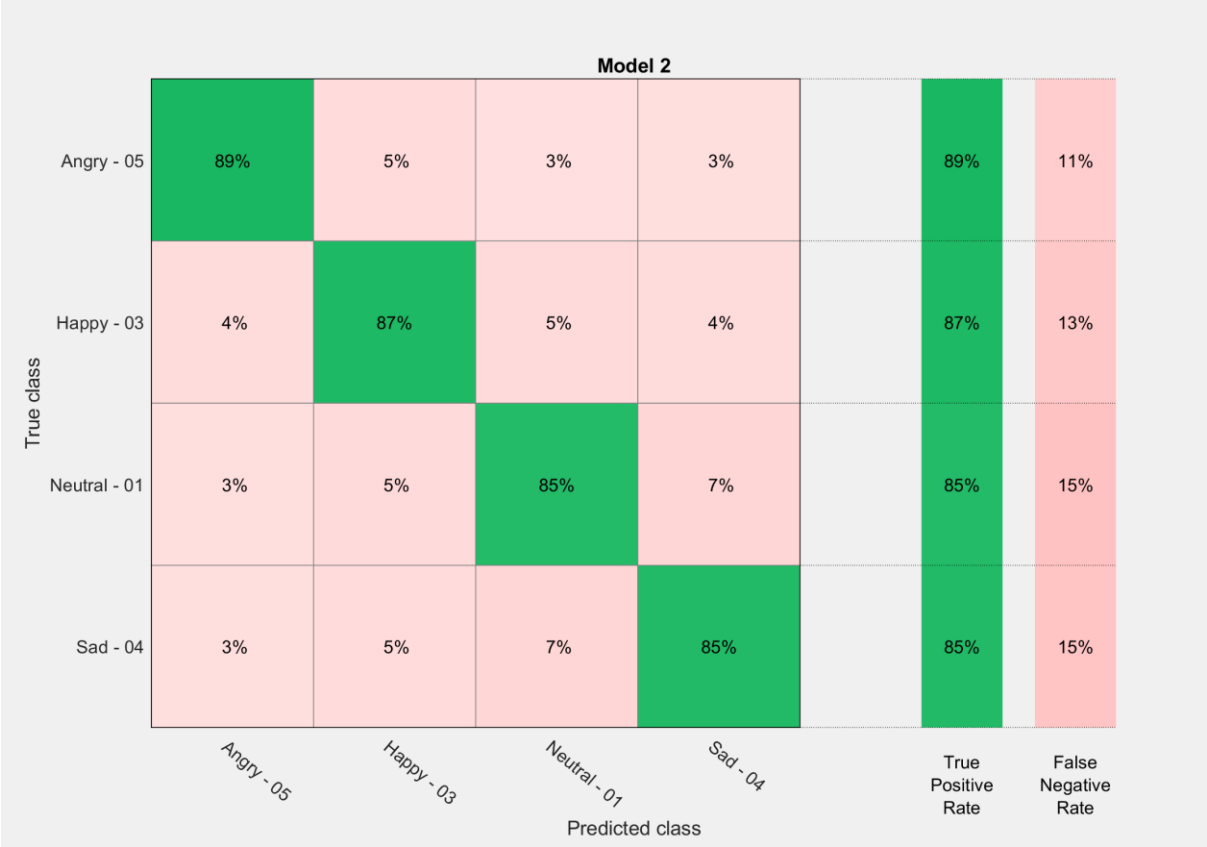

Fig. 5.6 Confusion Matrix of the True Positive rates and False Negative rates

Table 5.2 Emotion Recognition Rate with the KNN model

Emotions	Accuracy
Angry	89%
Happy	87%
Neutral	85%
Sad	85%

Overall, the KNN classification method for this data set is very accurate.

CHAPTER 6: Conclusion

Deduction of human emotions through voice and speech analysis has a practical plausibility and could potentially be beneficial for improving human conversational and persuasion skills. This paper presents a MATLAB approach for detection and analysis of human emotions on the basis of voice and speech processing. Two test cases have been examined, corresponding to four (4) emotional states – angry, happy, neutral and sad.

Each case demonstrates characteristics associated vocal features which can help in distinguishing the corresponding emotional state. Overall, the KNN classifier has a higher accuracy than the linear SVM classifier with close to 90% of the data points being classified correctly for each emotion class.

As future scope of work, more complex emotional states can be analysed such as fear, anxiety, disgust, boredom, frustration, etc. Also, databases with more diverse emotional speech can be used to achieve better results.

In order to improve on the original work of EmoVoice, 91699 observations for features were extracted for four (4) emotions which is far greater than the feature set of EmoVoice for 6 emotions. To improve the accuracy of detection, other features besides pitch and 13MFCC can be extracted. Some of these features include, but are not limited to signal entropy, spectral entropy, dominant frequency value, dominant frequency magnitude, etc.

References

[1] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C.M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan, “Analysis of emotion recognition using facial expressions, speech and multimodal information,” in Sixth International Conference on Multimodal Interfaces ICMI 2004, State College, PA, October 2004, pp. 205–211, ACM Press.

[2] Thurid Vogt, Elisabeth Andre et Nikolaus Bee, (2008). EmoVoice – A Framework for Online Recognition of Emotions from Voice

[3] Fei Tao, Gang Liu, Qingen Zhao, (2018). An Ensemble Framework of Voice-Based Emotion Recognition System for Films and TV Programs.

[4] S. Lalitha, A. Madhavan, B. Bhushan and S. Saketh, "Speech emotion recognition - IEEE Conference Publication", Ieeexplore.ieee.org. [Online]. Available: https://ieeexplore.ieee.org/iel7/6991454/7002373/07002390.pdf. [Accessed: 08- Dec- 2018].

[5] V. Nanavare and S. Jagtap, "Recognition of Human Emotions from Speech Processing", Procedia Computer Science, vol. 49, pp. 24-32, 2015. Available: 10.1016/j.procs.2015.04.223 [Accessed 9 December 2018].

[6] L. Devillers, L. Vidrascu, and L. Lamel, “Challenges in real-life emotion annotation and machine learning based detection,” Neural Networks, vol. 18, no. 4, pp. 407–422, June 2005.

[7] A. Batliner, K.Fischer, R.Huber, J.Spilker, and E. Nöth, “How to find trouble in communication,” Speech Communication, vol. 40, no. 1–2, pp. 117–143, April 2003.

[8] D. Litman, M. Rotaru, and G. Nicholas, “Classifying turn-level uncertainty using word level prosody,” in Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH), Brighton, UK, September 2009, pp. 2003–2006.

[9] G. Nicholas, M. Rotaru, and D. Litman, “Exploiting word-level features for emotion prediction,” in Proceedings of the IEEE/ACL Workshop on Spoken Language Technology, Palm Beach, Aruba, December 2006, pp. 110–113.

[10] D. Bitouk, A. Nenkova, and R. Verma, “Improving emotion recognition using class-level spectral features,” in Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH), Brighton, UK, September 2009, pp. 2023–2026.

[11] C. M. Lee, S. Yildirim, M. Bulut, and A. Kazemzadeh, “Emotion recognition based on phoneme classes,” in Proceedings of the 8th International Conference on Spoken Language Processing (INTERSPEECH), Jeju Island, Korea, October 2004, pp. 889–892.

[12] C. Busso, S. Lee, and S. Narayanan, “Using neutral speech models for emotional speech analysis,” in Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH), Antwerp, Belgium, August 2007, pp. 2225– 2228.

[13] B. Schuller, B. Vlasenko, D. Arsic, G. Rigoll, and A. Wendemuth, “Combining speech recognitionandacousticwordemotionmodelsforrobusttext-independentemotionrecognition,” in Proceedings of the IEEE International Conference on Multimedia & Expo (ICME), Hannover, Germany, June 2008, pp. 1333–1336.

[14] J. Kim, S. Lee, and S. Narayanan, “A detailed study of word-position effects on emotion expression in speech,” in Proceedings of the 10th Annual Conference of the

International Speech Communication Association (INTERSPEECH), Brighton, UK, September 2009, pp. 1987–1990.

[15] K. Rajvanshi, "An Efficient Approach for Emotion Detection from Speech Using Neural Networks", International Journal for Research in Applied Science and Engineering Technology, vol. 6, no. 5, pp. 1062-1065, 2018. Available: 10.22214/ijraset.2018.5170.

[16] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi, “Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques,” Journal of Computing, vol. 2, no. 3, pp. 138–143, Mar. 2010.

[17] G. Fink, “Developing HMM-based recognizers with ESMERALDA,” in Proceedings of the 2nd International Workshop on Text, Speech and Dialogue (TSD), Plzen, Czech Republic, September 1999, pp. 229–234.

[18] Johannes Wagner, Florian Lingenfelser, Tobias Baur, Ionut Damian, Felix Kistler, and Elisabeth André. 2013. The social signal interpretation (SSI) framework: multimodal signal processing and recognition in real-time. In Proceedings of the 21st ACM international conference on Multimedia (MM ’13). ACM, New York, NY, USA, 831-834. DOI=10.1145/2502081.2502223

[19] The MathWorks Inc. “Statistics and Machine Learning Toolbox: User's Guide” (R2019a). Retrieved March 6, 2019 from https://www.mathworks.com/help/audio/examples/speaker-identification-using-pitch-and-mfcc.html

[20] M.S. Likitha,1 Sri Raksha R. Gupta,2 K. Hasitha3 and A. Upendra Raju4 Dept. of Electronics, Mount Carmel College, Autonomous, Bangalore. Speech Based Human Emotion Recognition Using MFCC

[21] Dipti D. Joshi and M. B. Zalte, Recognition of emotion from marathi speech using MFCC and DWT algorithms,” International Journal of Advanced Computer Engineering

and Communication Technology (IJACECT), Issue-2, 2013, vol. 2, no. 2, pp. 59–63, 2013

[22] Shilna Sasheendran, M. J. Spoorthi, A. Upendra Raju, B. C. Shalini, and V. Uma, “Speaker identification and verification using MFCC and VQ methods,” International Journal of Scientific Engineering and Technology Research, vol. 3, no. 1, pp. 163–170, Jan. 2014

[23] "Supervised Learning Workflow and Algorithms- MATLAB & Simulink", Mathworks.com, 2019. [Online]. Available: https://www.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html#bswluh9. [Accessed: 19- Apr- 2019].

[24] Breiman, L. Random Forests. “Machine Learning 45”, 2001, pp. 5–32

[25] M. Khan, T. Goskula, M. Nasiruddin and R. Quazi, "Comparison between k-nn and svm method for speech emotion recognition", International Journal on Computer Science and Engineering, vol. 322011, no. 0975-3397, pp. 607-611, 2011. Available: https://www.researchgate.net/publication/50247425_Comparison_between_k-nn_and_svm_method_for_speech_emotion_recognition. [Accessed 23 April 2019].

[26] "KNN based emotion recognition system for isolated Marathi speech", International Journal of Computer Science Engineering, vol. 4, no. 2319-7323, pp. 173-177, 2015. Available: http://www.ijcse.net/docs/IJCSE15-04-04-076.pdf. [Accessed 23 April 2019].

[27] "Speaker Identification Using Pitch and MFCC- MATLAB & Simulink", Mathworks.com, 2019. [Online]. Available: https://www.mathworks.com/help/audio/examples/speaker-identification-using-pitch-and-mfcc.html. [Accessed: 05- Mar- 2019].

APPENDIX A

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Title: Speaker Identification Using Pitch and MFCC
%   Author: The Mathworks Inc
%   Date: 2019
%   Availability: https://www.mathworks.com/help/audio/examples/speaker-identification-
using-pitch-and-mfcc.html
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ADS = audioDatastore('D:\Maame Akua\Documents\2019 Spring
Semester\Capstone\Capstone
2019\Classification','IncludeSubfolders',true,'FileExtensions','.wav','LabelSource','foldernames')
;
[trainDatastore, testDatastore] = splitEachLabel(ADS,0.70);
trainDatastore; %displays the datastore
trainDatastoreCount = countEachLabel(trainDatastore); %displays the number of emotions in
the train datastore
%DatastoreCount = countEachLabel(ADS); %displays the number of emotions in the datastore
reset(trainDatastore);
%Feature Extraction
lenDataTrain = length(trainDatastore.Files); %finding the length of the traiining data
features = cell(lenDataTrain,1);
 for i = 1:lenDataTrain %starting an iteration/loop from 1 to the length of the training data
  [dataTrain, infoTrain] = read(trainDatastore);
  features{i} = HelperComputePitchAndMFCC(dataTrain,infoTrain); %extracting pitch and
MFCC features from each frame
 end
% features = vertcat(features{2:15});
features = vertcat(features{:});%vertical concatenation of all the features
features = rmmissing(features); %removing the rows or columns with missing entries
tail(features) %Display the last few rows
featureVectors = features{:,2:15};
m = mean(featureVectors); %finding the mean
s = std(featureVectors); %finding the standard deviation
features{:,2:15} = (featureVectors-m)./s; %normalizing the features
tail(features)
%testing the data
lenDataTest = length(testDatastore.Files);
featuresTest = cell(lenDataTest,1);
for j = 1:lenDataTest
  [dataTest, infoTest] = read(testDatastore);
  featuresTest{j} = HelperComputePitchAndMFCC(dataTest,infoTest);
end
```

```
featuresTest = vertcat(featuresTest{:});
featuresTest = rmmissing(featuresTest);
featuresTest{:,2:15} = (featuresTest{:,2:15}-m)./s;
head(featuresTest)   % Display the first few rows
```
[2]