Speech emotion recognition system

A PROJECT REPORT

Submitted by

Anant Dev Pandey (20BCS9366)

Alisha Chhabra (20BCS9340

Rohit Garg (20BCS9368)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

Chandigarh University

MAY 2022

BONAFIDE CERTIFICATE

Certified that this project report **" Speech emotion recognition system"** is the bonafide work of "**Anant Dev Pandey, Alisha Chhabra, Rohit Garg"** who carried out the project work under my/our supervision.

SIGNATURE

Dr. Neeraj
SUPERVISOR

SIGNATURE

Dr. Puneet Kumar
HEAD OF THE DEPARTMENT

Submitted for the project viva-voce examination held on_____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our respected and learned guides, **Dr. Neeraj** for their valuable help and guidance, we are thankful to them for the encouragement they have given us in completing the project.

We are also grateful to our respected HOD **Dr. Puneet Kumar** and all other teachers for providing us assistance in various stages during the course of our project.

Signature of all the group members

1. Anant Dev Pandey

2. Alisha Chhabra

3. Rohit Garg

CONTENT

Abstract 6

1. Introduction 7
1.1 Overview 7
1.2 Objectives *8*
1.3 Literature Review *8*

2. System Analysis 9

2.1 Identification of Need 9

2.2 Preliminary Investigation 10

2.3 Feasibility Study 10
2.3.1 Technical Feasibility 11
2.3.2 Economic Feasibility 11
2.3.3 Schedule Feasibility 12
2.3.4 Legal Feasibility 13

2.4 Project Planning 13

2.5 Project Scheduling 14

2.6 Constraints, Assumptions and Dependencies *15*

2.7 Software Engineering Paradigm 15
2.7.1 Process 17
2.7.2 Advantages 18

2.8 Data Models and Description *20*
2.8.1 Sequence Diagram *20*
2.8.2 ER Diagram *21*
2.8.3 Flowchart *22*

3. Sequence Diagram *23*
3.1 Implementation 23
3.2 User Interface 24

4. Coding 27
4.1 Coding Standardization 27
4.2 Source Code 28

5. Conclusion *31*

References 32

Figure Index

Figure 1. Block diagram of proposed system……..... 17
Figure 2. Schema of Mfcc extraction …….……. *20*
Figure 3. Process of computing the ST representation…… *21*
Figure 4. Basic concept of RNN……... *22*
Figure 5. Iterative model of software development…….. *24*
Figure 6. Data flow diagram.…….…….……... *25*
Figure 7. Flowchart…….. *25*
Figure 8. ER diagram……... *26*
Figure 9. Home page ……..…….. *28*
Figure 10. After recording voice …….…… *28*
Figure 11. Code Screenshot.……. *29*

ABSTRACT

As of late, the significance of responding to the emotional condition of a client has been by and large acknowledged in the field of human-computer cooperation and particularly speech has gotten expanded center as a methodology from which to deduct data on feeling consequently. Up to this point, mostly scholarly furthermore, not very application-arranged disconnected investigations in light of recently recorded and clarified data sets with emotional speech were led. In any case, requests of online examination vary from that of disconnected examination, specifically, conditions are seriously difficult and less unsurprising.

In the field of speech emotion recognition numerous strategies have been used to extricate emotions from signals, counting some deep rooted speech examination and grouping methods. In the customary method of speech emotion recognition highlights are removed from the speech signals and afterward the highlights are chosen which is all in all know as determination module and afterward the emotions are perceived this is an exceptionally extensive and time taking interaction so this paper gives an outline of the profound learning strategy which depends on a straightforward calculation in light of component extraction and model creation which perceives the emotion.

The remainder of the article is coordinated as follows*s 1) Introduction, 2) Literature Survey, 3) Design Flow \ Process, 4) Result, 5) Conclusion\Future work, 6) Reference.*

1. INTRODUCTION

1.1 Overview

Emotion assumes a critical part in everyday relational human communications. This is fundamental for our objective as well as keen choices. It assists us with coordinating and figure out the sensations of others by passing our sentiments and giving criticism on to other people. Research plays uncovered the strong part that emotion play in forming human social communication. Emotional presentations pass on impressive data about the psychological condition of a person. This has opened up another examination field called programmed emotion recognition, having essential objectives to comprehend and recover wanted emotions.

Investigations of programmed emotion recognition frameworks mean to make effective, ongoing techniques for recognizing the emotions of cell phone clients, call focus administrators and clients, vehicle drivers, pilots, and numerous other human-machine correspondence clients. Adding emotions to machines has been perceived as a basic consider causing machines to show up and act in a human-like way Robots equipped for understanding emotions could give fitting emotional reactions and display emotional characters. In certain conditions, people could be supplanted by computer-created characters being able to lead extremely normal and persuading discussions by engaging human emotions. Machines need to comprehend emotions conveyed by speech. Just with this ability, a totally significant discourse in light of common human-machine trust and understanding can be accomplished.

1.2 Objectives

This article took a gander at how you can involve speech information in certifiable applications, including programmed speech recognition (ASR) and speech emotion recognition (SER). We investigated opensource Python bundles to assist with beginning ASR and recommended project thoughts. We likewise brought a more profound jump into building a strong SER model utilizing the TESS (TORONTO EMOTION SPEECH SET) dataset to prepare a LSTM model. This active experience will prepare you to begin building undertakings and expert the ideas of SER.

Researchers apply different sound handling procedures to catch this secret layer of data that can enhance and concentrate apparent and acoustic highlights from speech Changing over sound signs into numeric or vector design isn't quite so direct as pictures. The change technique will decide how much essential data is held when we leave the "sound" design. In the event that a specific information change can't catch the delicateness and serenity, it would be trying for the models to gain proficiency with the emotion and order the example.

A few techniques to change sound information into numeric incorporate Mel Spectrograms that imagine sound signs in light of their recurrence parts which can be plotted as a sound wave and took care of to prepare a CNN as a picture classifier. We can catch this utilizing Mel-recurrence cepstral coefficients (MFCCs). Every one of these information designs has its advantages and impediments in view of the application. We will attempt to get the information from the MFCC and plot the information in a reasonable cluster structure that is involved by the model for instance we are utilizing here the LSTM model of element recognition we will utilize the numeric qualities given by the MFCC as contribution to the LSTM model and will attempt to perceive the emotion.

1.3 Literature Review

Complete survey on the speech emotion recognition is made sense of in [9] which audits properties of dataset, speech emotion recognition concentrate on classifier decision. Different acoustic highlights of speech are explored and a portion of the classifier strategies are dissected in [10] which is useful in the further examination of current techniques for emotion recognition. This paper [11] explored the forecast of the following responses from emotional vocal signs in light of the recognition of emotions, utilizing various classifications of classifiers. A portion of the classification algorithms like K-NN, Random Forest are utilized in [11] to as needs be group emotion. Repetitive Brain network emerges hugely which attempts to tackle numerous issues in the recorded of information science. Deep RNN like LSTM, Bi-directional LSTM prepared for acoustic highlights are utilized in [12]. Different scope of CNN are being executed and prepared for speech emotion recognition are assessed in [13]. Emotion is induced from speech signals utilizing channel banks and Deep CNN[14] which shows high exactness rate which gives a derivation that deep learning can likewise be utilized for emotion identification. Speech emotion recognition can be additionally performed utilizing picture spectrograms with deep convolutional networks which is carried out in [15].

2. SYSTEM ANALYSIS

2.1 Identification of Need

Identification of need is a process of determining what and how an end-user would expect a product to perform after the deployment at production level. There's also non- technical needs of an end-user or a business client which reflects the users' perception of the product and not the actual technical workaround, but they are closely related to the technical need at times. By implementing a needs identification system, the organization helps to ensure the proper allocation of assets to different project within the organization.

Identifying Problems

Identifying potential problems before the start of a project can save the organization significant amounts of time and money. Problem analysis is one of the most critical stages of project planning because this stage helps to guide all subsequent analysis and decision-making. If the project does not advance past this stage with solutions that the organization can implement, the project should not go forward in its current form.

Observations

The needs for a project are identified after the organization makes observations about the project. Observations are often subjective and therefore someone with expertise about the proposed project should help to make observations. A good observer can identify the needs of the project by answering key questions about the project. If the observations take into consideration the project itself and the outcome of the project, the observations should meet all of the needs of the project.

Gathering Information

Observation and gathering information represent two processes. Observations highlight what is needed. On the other hand, gathering information highlights the processes needed to execute the proposed project. Both observations and the actual gathering of information should include comments from the group that ultimately will benefit from the completed project.

Objectives and Opportunities

Once the organization has analyzed the needs and identified the objectives, the organization needs to allocate funds to capitalize the project. By successfully identifying the needs, an organization can begin to allocate resources to pay for the project. Additionally, a business needs to consider the potential future cash flow of the project. This allows the business to analyze potential cost savings to minimize costs and maximize the efficiency of the project.

2.2 Preliminary Investigation

The main aim of preliminary investigation is to identify the problem. First, need for the new or the enhanced system is established. Only after the recognition of need, then the proposed system is compared and then further analysis is possible. At this stage, we had to perceive the problem and opportunities, the existing system is studied and found out that there were few areas where we can integrate with other technology to make the system better than the existing system. It was analyzed that such proposed system would be possible to develop with given and it might turn out to be the feasible solution.

In this project, the biggest challenge was to integrate the existing social media with the designed framework and on further development levels we encountered various unit level problems such as the model for the secure transmission of content. On the later part of this document, we have come up the features which can be added to our software to make it better than the initial deployment.

2.3 Feasibility Study

A feasibility study is a high-level capsule version of the entire system analysis and design process. The study begins by classifying the problem definition. The purpose of feasibility study is not to solve the problem, but to determine whether the problem is worth solving. It is a preliminary study which is conducted before the real development of the project commences not keeping the factor of project's success. It creates a roadmap of what are the possible solutions if we choose a certain path. The feasibility study concentrates on the following areas:

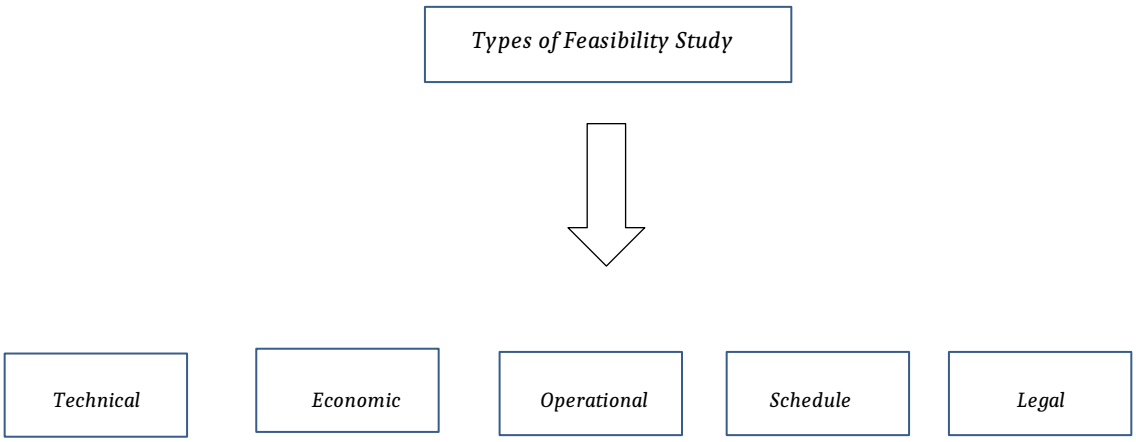


Fig. A: Types of Feasibility Study

2.3.1 Technical Feasibility

Evaluating the technical feasibility study is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed designs of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis. Understand the different technologies involved in the proposed system before commencing the project we have to be very clear about what are the technologies that are to be required for the development of the new system. Overall, this study needs to demonstrate that the proposed system which is need to be developed is technically feasible.

This requires:

- An outline of the requirements,
- A possible system design,
- Possible choices of software to be used or developed,
- Estimates on number of users, data, etc.

2.3.2 Economic Feasibility

The economic feasibility study evaluates the cost of the software development against

the ultimate income or benefits gets from the developed system. There must be scopes for profit after the successful Completion of the project. The life cycle of an engineering project or product contains of several stages, namely: (i) Planning and design; (ii) Development; (iii) Operation and maintenance. It should be performed to identify the financial risk associated with the project.

Various techniques like net present value (NPV), payback period, return on investment (ROI) are employed. Techno-Economic Assessment (TEA) is a cost-benefit comparison using different methods. These assessments are used for tasks such as:

- Evaluate the economic feasibility of a project.
- Investigate cash flows over the lifetime of the project.
- Evaluate the likelihood of different technology scales and applications.
- Compare the economic quality of different technology application providing the same service.

2.3.3 Schedule Feasibility

It means that the project can be implemented in an acceptable time frame. When assessing schedule feasibility, a systems analyst must consider the interaction between time and costs. For example, speeding up a project schedule might make a project feasible, but much more expensive.

Other issues that relate to schedule feasibility include the following:

- Can the company control the factors that affect schedule feasibility?
- Has management established a firm timetable for the project?
- What conditions must be satisfied during the development of the system?
- Will an accelerated schedule pose any risks? If so, are the risks acceptable?
- Will project management techniques be available to coordinate and control the project?
- Will a project manager be appointed?

It is also the likelihood that timeframes can be met and that this is adequate to meet organization’s needs.

2.3.4 Legal Feasibility

It determines whether the proposed system conflicts with the legal requirements, in this case as we didn't try to execute anything on the public domain, hence this project is legal feasible.

It is important that the project is following the requirements needed to start a project including certificates, copyrights, business insurance, tax number, health and safety measures and many more. There are some things to consider in legal feasibility study including ethical issues and some social issues. These issues are the privacy and accountability. In this project, everything is designed keeping in mind all the legal terms and no real-world data or privacy has been breached of any person of this country to use it as a sample user to implement this application.

2.4 Project Planning

Our SER framework comprises of four primary advances. First is the voice test assortment. The second elements vector that is shaped by removing the highlights. As the following stage, we attempted to figure out which highlights are generally pertinent to separate every emotion. These highlights are acquainted with machine learning classifier for recognition. This cycle is depicted in Figure 1.

Feature extraction

The speech signal contains a large number of parameters that reflect the emotional characteristics. One of the sticking points in emotion recognition is what features should be used. In recent research, many common features are extracted, such as energy, pitch, formant, and some spectrum features such as linear prediction coefficients (LPC), mel-frequency cepstrum coefficients (MFCC), and modulation spectral features. In this work, we have selected modulation spectral features and MFCC, to extract the emotional features. Mel-frequency cepstrum coefficient (MFCC) is the most used representation of the spectral property of voice signals. These are the best for speech recognition as it takes human perception sensitivity with respect to frequencies into consideration. For each frame, the Fourier transform and the energy spectrum were estimated and mapped into the Mel-frequency scale. The discrete cosine transform (DCT)

of the Mel log energies was estimated, and the first 12 DCT coefficients provided the

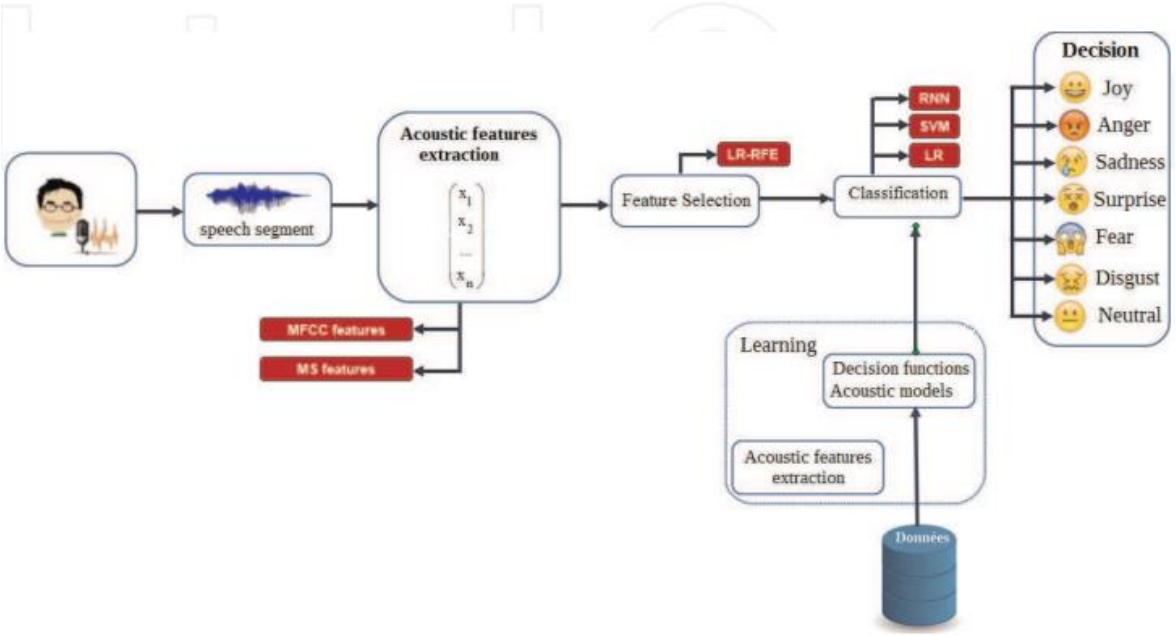


Figure 1.
Block diagram of the proposed system.

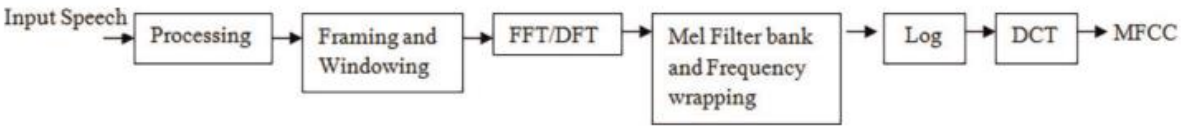


Figure 2.
Schema of MFCC extraction [33].

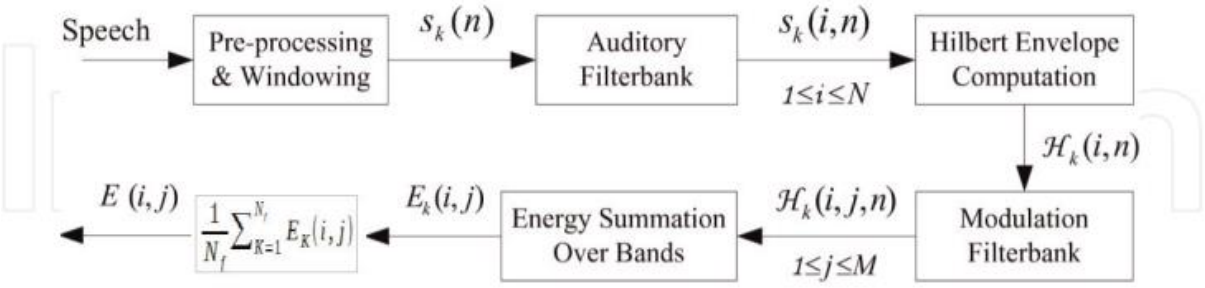


Figure 3.
Process for computing the ST representation [5].

MFCC values used in the classification process. Usually, the process of calculating MFCC is shown in Figure 2. In our research, we extract the first 12 order of the MFCC coefficients where the speech signals are sampled at 16 KHz. For each order coefficients, we calculate the mean, variance, standard deviation, kurtosis, and skewness, and this is for the other all the frames of an utterance. Each MFCC feature vector

is 60-dimensional.

Feature selection

As reported by Aha and Bankert, the objective of feature selection in ML is to "reduce the number of features used to characterize a dataset so as to improve a learning algorithm's performance on a given task." The objective will be the maximization of the classification accuracy in a specific task for a certain learning algorithm; as a collateral effect, the number of features to induce the final classification model will be reduced. Feature selection (FS) aims to choose a subset of the relevant features from the original ones according to certain relevance evaluation criterion, which usually leads to higher recognition accuracy. It can drastically reduce the running time of the learning algorithms. In this section, we present an effective feature selection method used in our work, named recursive feature elimination with linear regression (LR-RFE). Recursive feature elimination (RFE) uses a model (e.g., linear regression or SVM) to select either the best- or worst-performing feature and then excludes this feature. These estimators assign weights to features (e.g., the coefficients of a linear model), so the goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features, and the predictive power of each feature is measured. Then, the least important features are removed from the current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached. In this work, we implemented the recursive feature elimination method of feature ranking via the use of basic linear regression (LR-RFE). Other research also uses RFE with another linear model such as SVM-RFE that is an SVM-based feature selection algorithm created by. Using SVM-RFE, Guyon et al. selected key and important feature sets. In addition to improving the classification accuracy rate, it can reduce classification computational time.

Classification methods

Many machine learning algorithms have been used for discrete emotion classification. The goal of these algorithms is to learn from the training samples and then use this learning to classify new observation. In fact, there is no definitive answer to the choice of the learning algorithm; every technique has its own advantages and limitations. For this reason, here we chose to compare the performance of three different classifiers. Multivariate linear regression classification (MLR) is a simple and efficient computation of machine learning algorithms, and it can be used for both regression and classification problems. We have slightly modified the LRC algorithm described as follow Algorithm 1. We calculated (in step 3) the

absolute value of the difference between original and predicted response vectors ($|y - y_i|$), instead of the Euclidean distance between them ($||y - y_i||$). Support vector machines (SVM) are an optimal margin classifier in machine learning. It is also used extensively in many studies that related to audio emotion recognition which can be found in [10, 13, 14]. It can have a very good classification performance compared to other classifiers especially for limited training data. SVM theoretical background can be found in. A MATLAB toolbox implementing SVM is freely available in. A polynomial kernel is investigated in this work

Recurrent neural networks (RNN) are suitable for learning time series data, and it has shown improved performance for classification task. While RNN

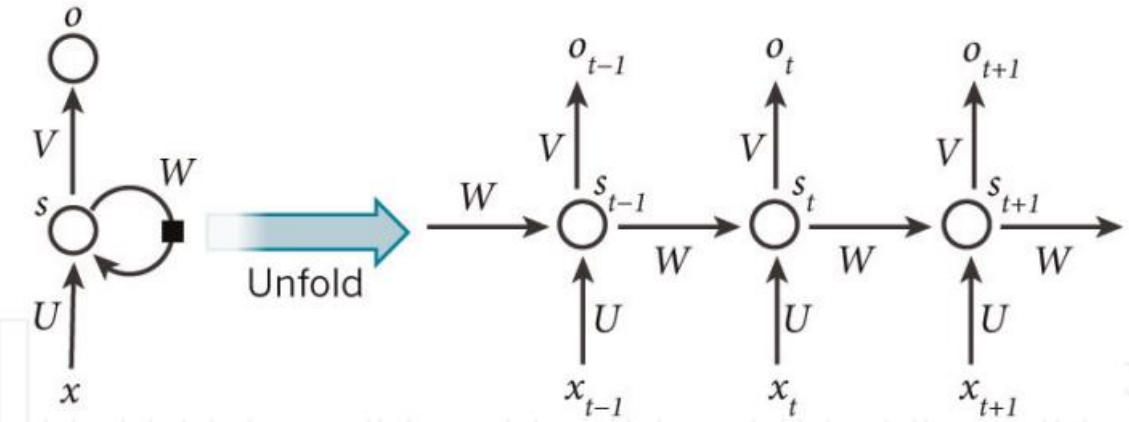


Figure 4.
A basic concept of RNN and unfolding in time of the computation involved in its forward computation [18].

models are effective at learning temporal correlations, they suffer from the vanishing gradient problem which increases with the length of the training sequences. To resolve this problem, long short-term memory (LSTM) RNNs were proposed by Hochreiter et al. it uses memory cells to store information so that it can exploit long-range dependencies in the data. Figure 4 shows a basic concept of RNN implementation. Unlike traditional neural network that uses different parameters at each layer, the RNN shares the same parameters (U, V, and W are presented in Figure 4) across all steps. The hidden state formulas and variables are as follows:

$$s_t = f\,(Ux_i + Ws_{t-1})$$

where x_t, s_t, and o_t are respectively the input, the hidden state, and the output at time step t and U,V,W are parameters matrices.

2.5 Project Scheduling

It requires us to follow some carefully laid-out steps, in order, for the schedule to take shape. It is an organized method of presenting information on when activities need to be started, how long activities are planned to be completed.

There are basic principles for project scheduling, such as follows:

- **Defined responsibilities**
  - Every task that is scheduled is assigned tzo a specific team member.

- **Defined outcomes**
  - Every task that is scheduled should have a defined outcome for software projects such as a work product.
- **Define milestones**
  - Every task or group of tasks should be associated with a project milestone.
  - A milestone is accomplished when one or more work products has been reviewed and then approved by the team leader.

2.6 Software Engineering Paradigm

This project uses an iterative model approach using Agile methodologies. Let's discuss this in details. Agile methods of software development are most commonly described as iterative and incremental development. The iterative strategy is the cornerstone of Agile practices, most prominent of which are SCRUM, DSDM, and FDD. The general idea is to split the development of the software into sequences of repeated cycles (iterations). Each iteration is issued a fixed-length of time known as a timebox. A single timebox typically lasts 2-4 weeks.

The ADCT (Analysis, Design, Code, Test) wheel is more technically referred to as the PDCA (Plan, Design, Check, Adjust) cycle. The team implements the PDCA cycle on each iteration separately in the following manner:

- **P (Plan) – Iteration Planning**

In this event, the team collaborates to discuss the objectives for the next iteration. It also

summarizes the work done and determines the team backlog required for the next iteration.

- **D (Design) – Iteration Execution**

This is the ‘do’ step where the development of the software, its design and coding takes place. If it’s a second or third iteration, then functionality testing is also conducted. The team collects user stories and prepares for the next step, that is the Iteration Review.

- **C (Check) – Iteration Review**

Also known as the ‘check’ step, Iteration Review is carried out with the Product Owner. The team shows the tested deliverable to the Product Owner, who then reviews the completed work and ascertains whether all criteria have been met.

- **A (Adjust) – Iteration Retrospect**

In this event, the team evaluates the entire process of the iteration from the first step. It essentially works on any improvements that are gathered in previous iterations. New problems are identified along with their causes. Before the team starts the next cycle again, team backlog is refined for future reference. The iterations are repeated for optimizations and improvisations and, the lessons learned from previous cycles are applied in the next cycle. Until a fully functional software is ready to hit the market.

Agile methodologies have the following advantages over other methods:

Customer Involvement – Agile Iterative development encourages user contribution. After each iterative cycle, customer feedback is obtained, and the product is then subjected to necessary changes based on that feedback. This aspect brings adaptability into the project’s framework.

Risk Assessment – Agile iteration allows risk identification and mitigation early on in the development to avoid speed bumps later down the timeline.

Rapid Delivery – The work is divided into small cycles, allowing team members to dedicate their focus and deliver on time. Moreover, testing is conducted simultaneously in coding and design in every iteration, which greatly reduces the time needed to achieve completion.

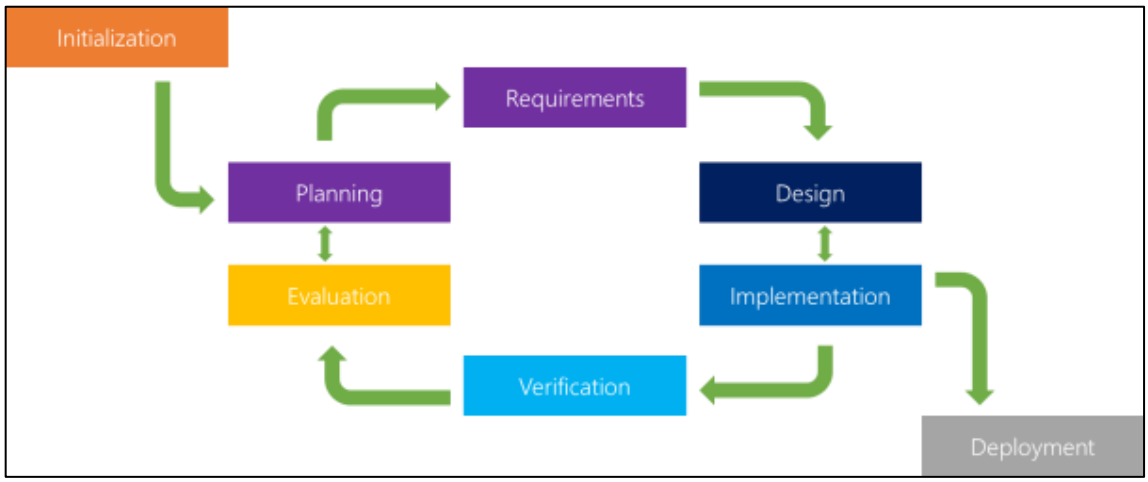


Figure 5: Iterative Model of Software Development

2.6.1 Process

- **Planning & Requirements**: As with most any development project, the first step is to go through an initial planning stage to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle.

- **Analysis & Design**: Once planning is complete, an analysis is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The design stage also occurs here, establishing any technical requirements (languages, data layers, services, etc.) that will be utilized in order to meet the needs of the analysis stage.

- **Implementation**: With the planning and analysis out of the way, the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.

- **Testing**: Once this current build iteration has been coded and implemented, the next step is to go through a series of testing procedures to identify and locate

any potential bugs or issues that have cropped up.

- **Evaluation**: Once all prior stages have been completed, it is time for a thorough evaluation of development up to this stage. This allows the entire team, as well as clients or other outside parties, to examine where the project is at, where it needs to be, what can or should change, and so on.

2.7 Data Models and Descriptions

2.7.1 Data flow Diagram

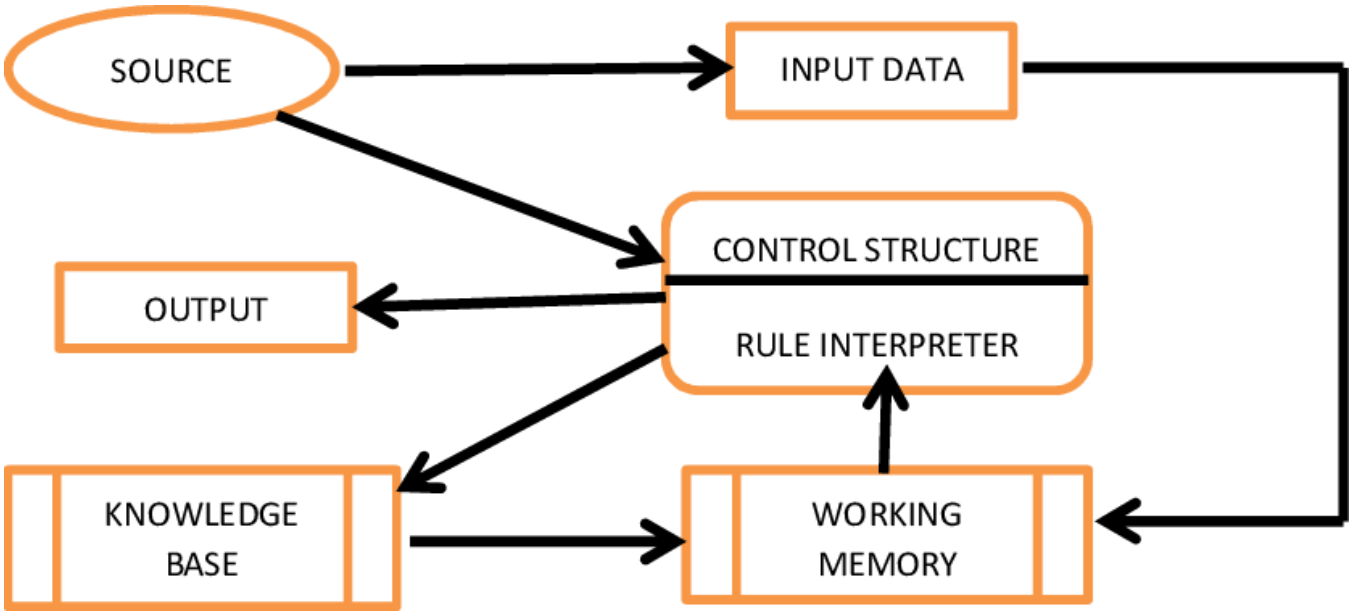


Figure 6: Data flow Diagram

2.7.2 Entity Relationship Diagram

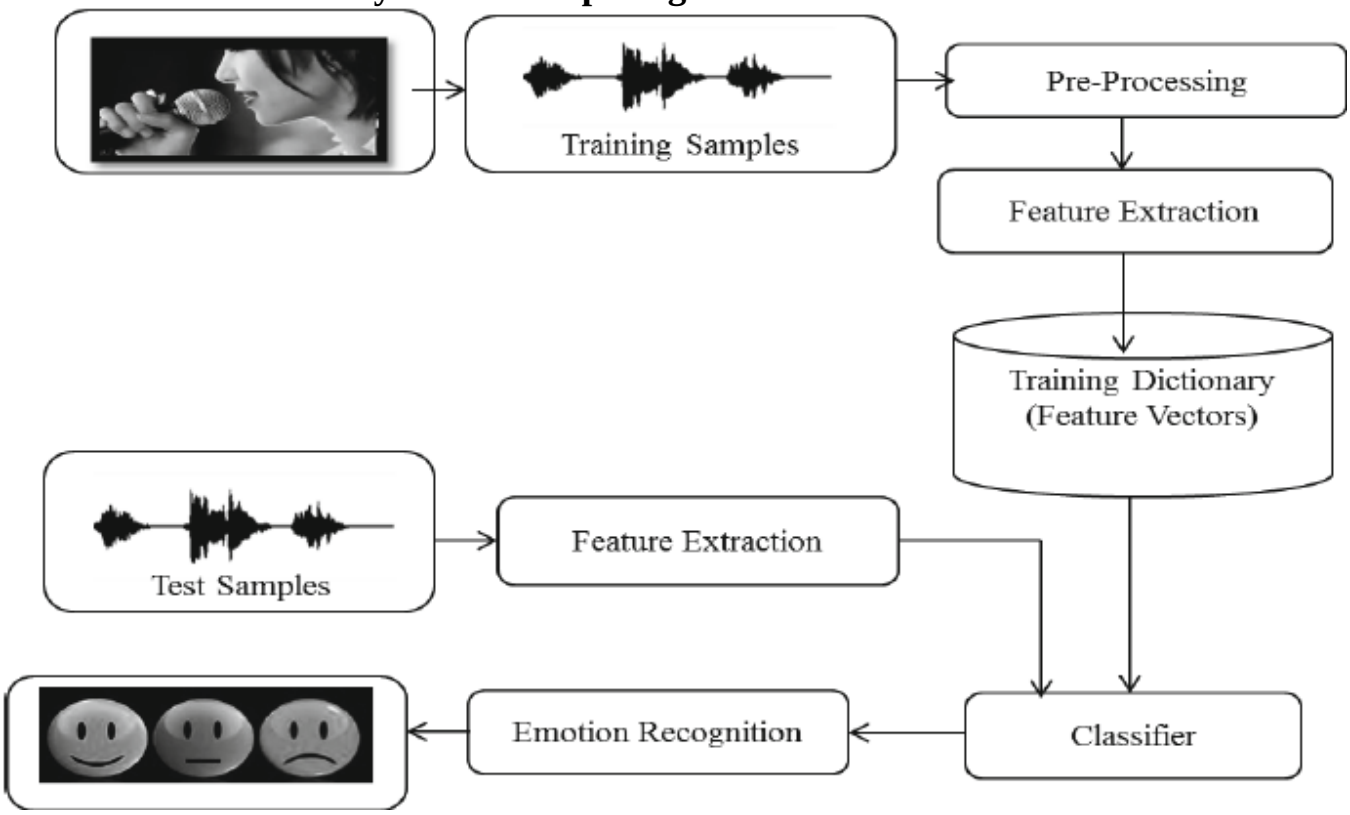


Figure 7: ER Diagram

2.7.3 Flowchart

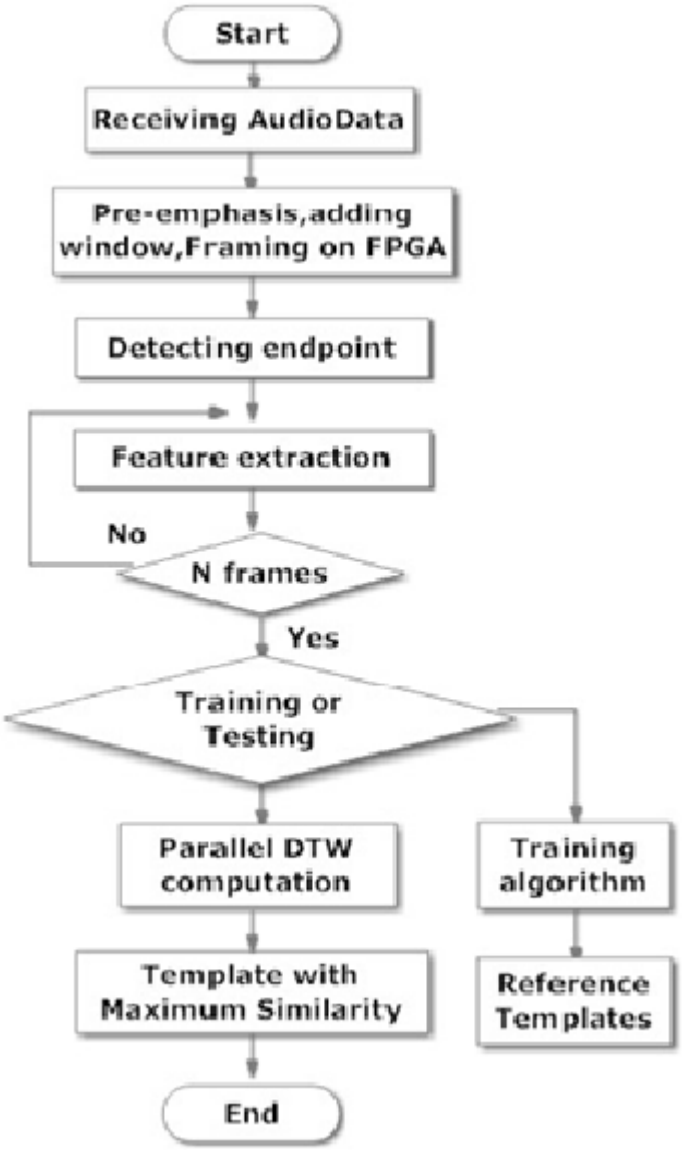


Figure 8: Flowchart

SYSTEM DESIGN

*3.1*Data collection

The first step in implementing the Speech Emotion Recognition system is to collect audio samples under different emotional categories which can be used to train the model. The audio samples are usually wav or mp3 files and publically available for download. The following steps are explained relative to the experiments performed on the TESS dataset.

3.2 Python library

The next step after data collection was to represent these audio files numerically, in order to perform further analysis on them. This step is called feature extraction, where quantitative values for different features of the audio is obtained. The pyAudioAnalysis library was used for this purpose [15]. This python library provides functions for short-term feature extraction, with tunable windowing parameters such as frame size and frame step. At the end of this step, each audio file was represented as a row in a CSV file with 34 columns representing the different features. Each feature will have a range of values for one audio file obtained over the various frames in that audio signal. The python library pyAudioAnalysis is an open Python library that provides a wide range of audio-related functionalities focusing on feature extraction, classification, segmentation, and visualization issues. The library depends on several other libraries which are:

- Numpy
- Matplotlib
- Scipy
- Sklearn
- Librosa

2) User Management Contract

The UMC uses two data structures to store the friend information list and the post list. The friend information consists of the friend identity and status. The status is an

enumeration variable with three optional values: pending, applying, and approval. When a user is in a pending or approval state for a friend, he can query this friend's information in the URC by using the friend identity. In order to reduce the data size stored in the, the post list only keeps hash values of all posts and their corresponding timestamps. In addition to the aforementioned data stored in the UMC, there are some interfaces and corresponding events in the following.

3.3 Feature engineering and model analysis

Feature engineering is the process of transforming, reducing or constructing features for the dataset. As mentioned earlier in the raw data, each feature has multiple values for each frame of the audio signal. By the frame blocking and windowing techniques, the frame size and frame overlap values can be tuned to obtain accurate values of the audio signal. Further, using the averaging technique, average values of different features for the audio signals are obtained.

For creating the LSTM model we have to import the sequential, dense, LSTM and the dropout form keras.models and keras.layers respectively. Then specifying the various values required. In the next step compiling the model for loss and accuracy and then model.summary() to view the model it

User Interface and code

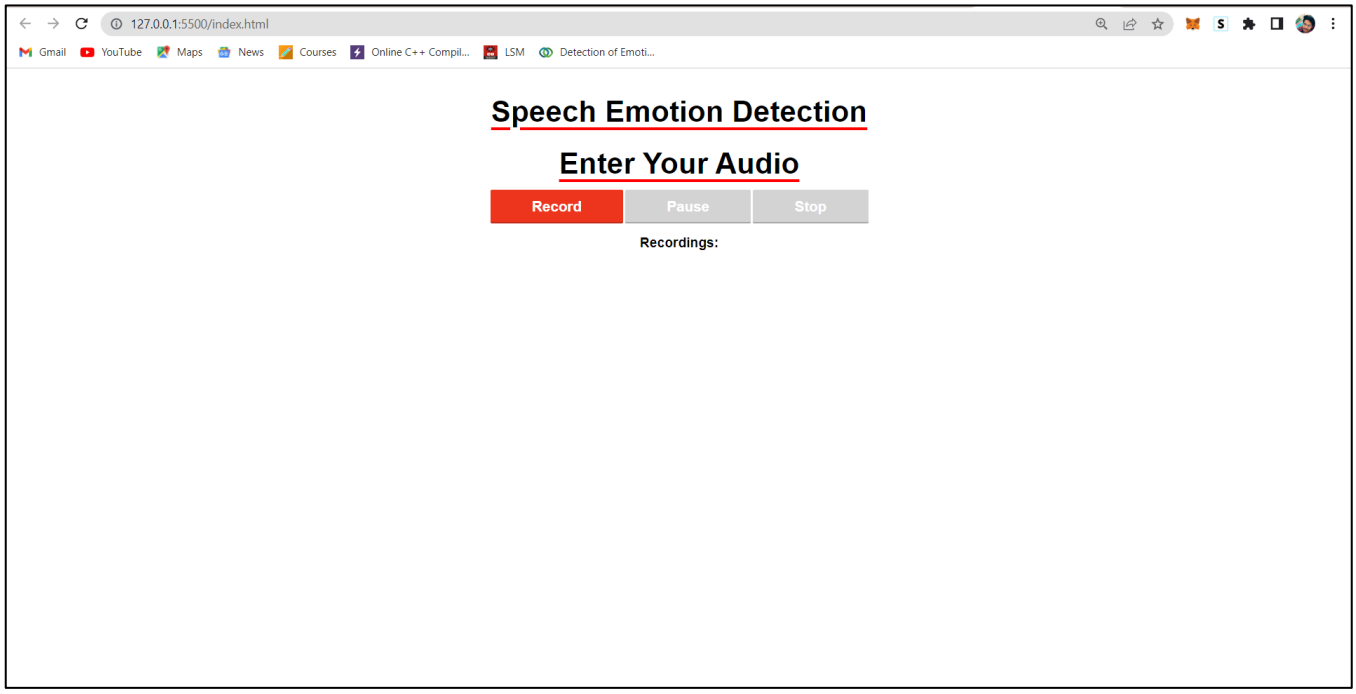


Figure 9: Homepage

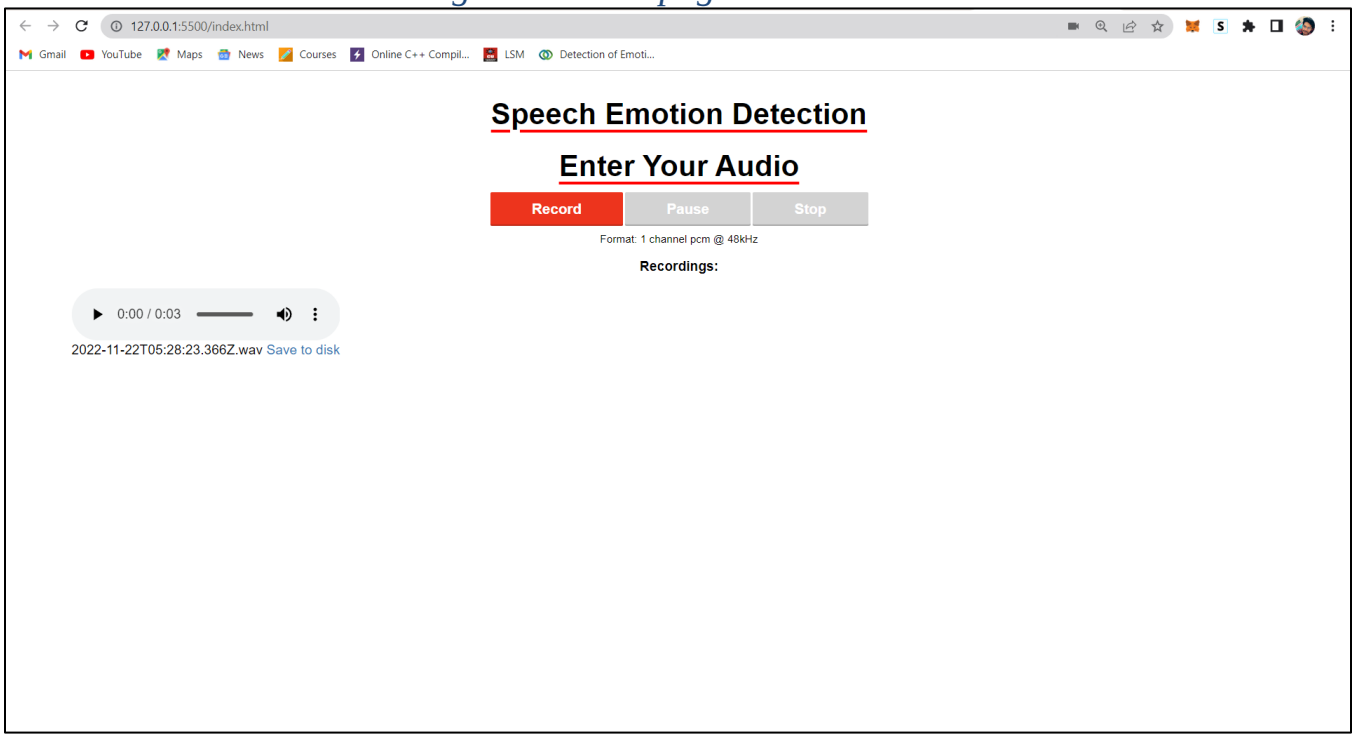


Figure 10: After recording voice

3. CODING

The above system design is translated into a machine-readable form which is termed as coding. It is basically translating the human readable format to a machine friendly one. The code generation step performs this task.

The following points are considered while converting the system design into coding:

- Are the initializations correct?
- Are the data types properly assigned?
- Is memory leak being dealt with?
- Does it comply with the coding standard?

4.1 Coding Standardization

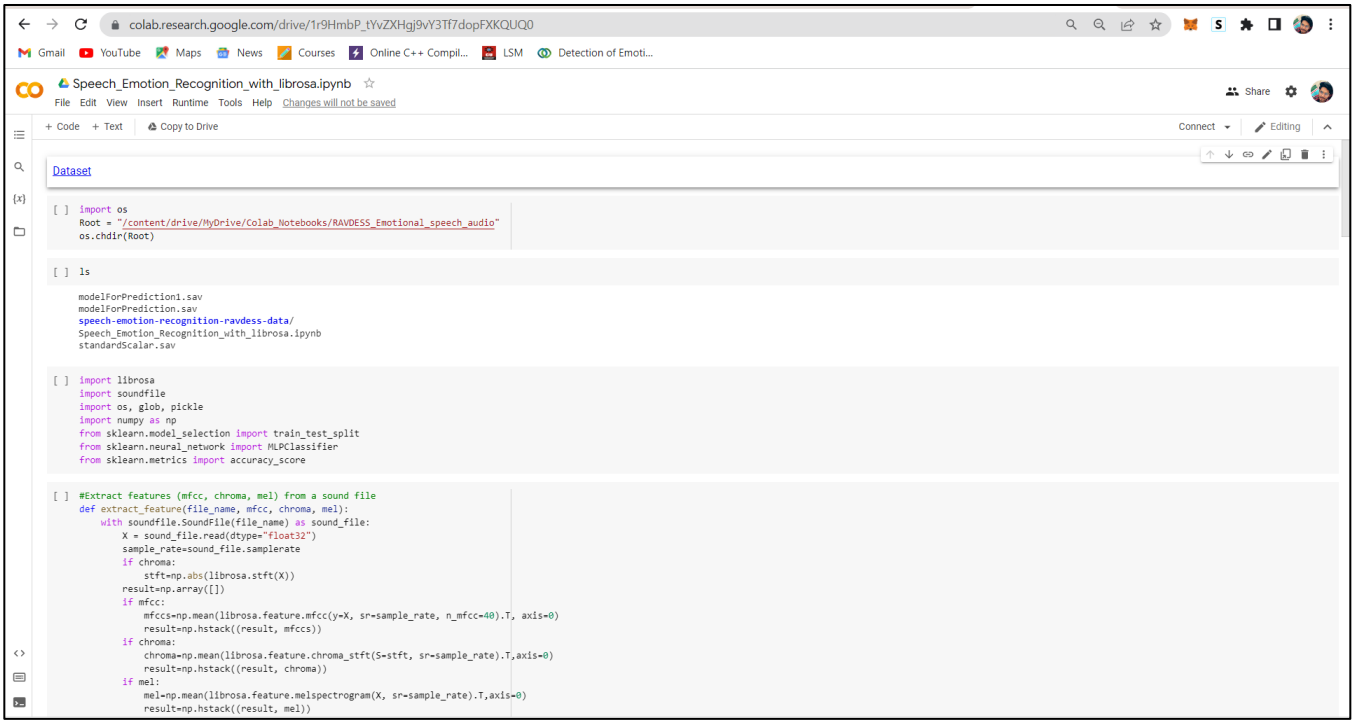
Coding Standardization basically the efficiency of our code which has been converted from the system design. The efficiency primarily depends upon:

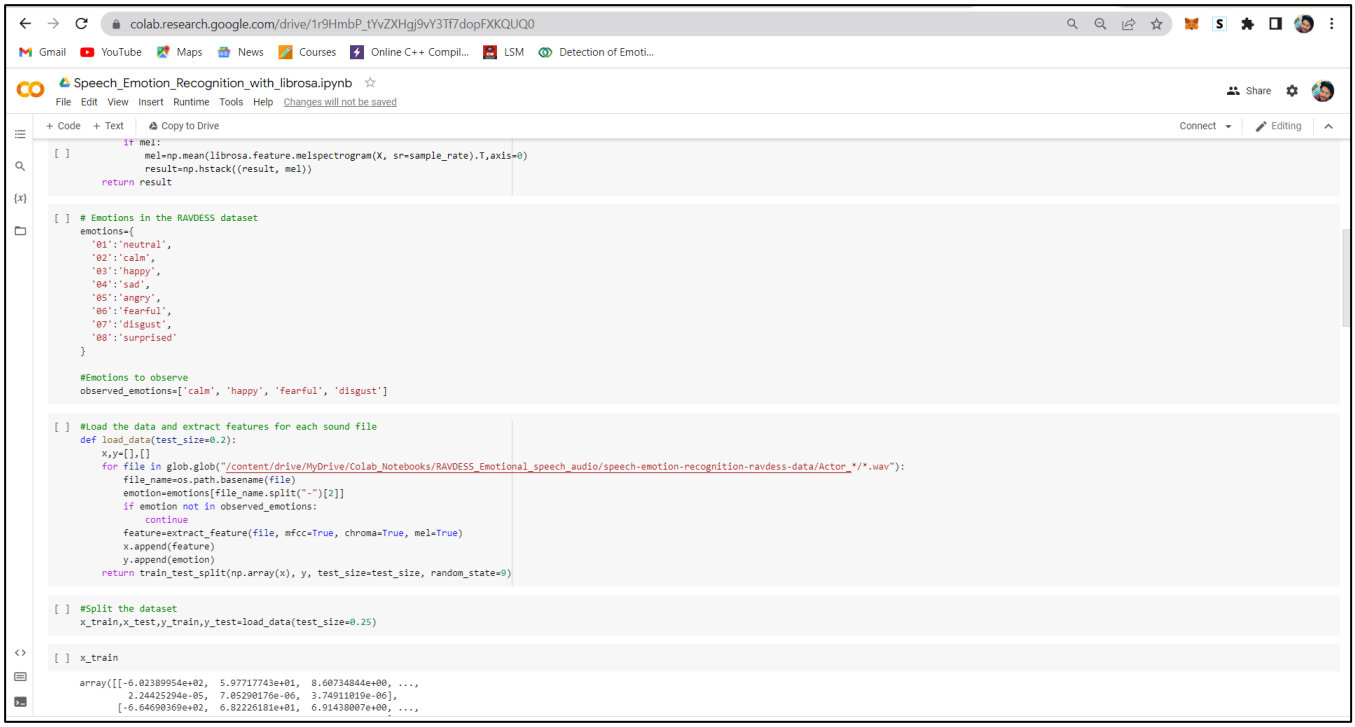
- *Readability*: The code should be readable with proper indentation and spacing to make the contents clear of all the modules.

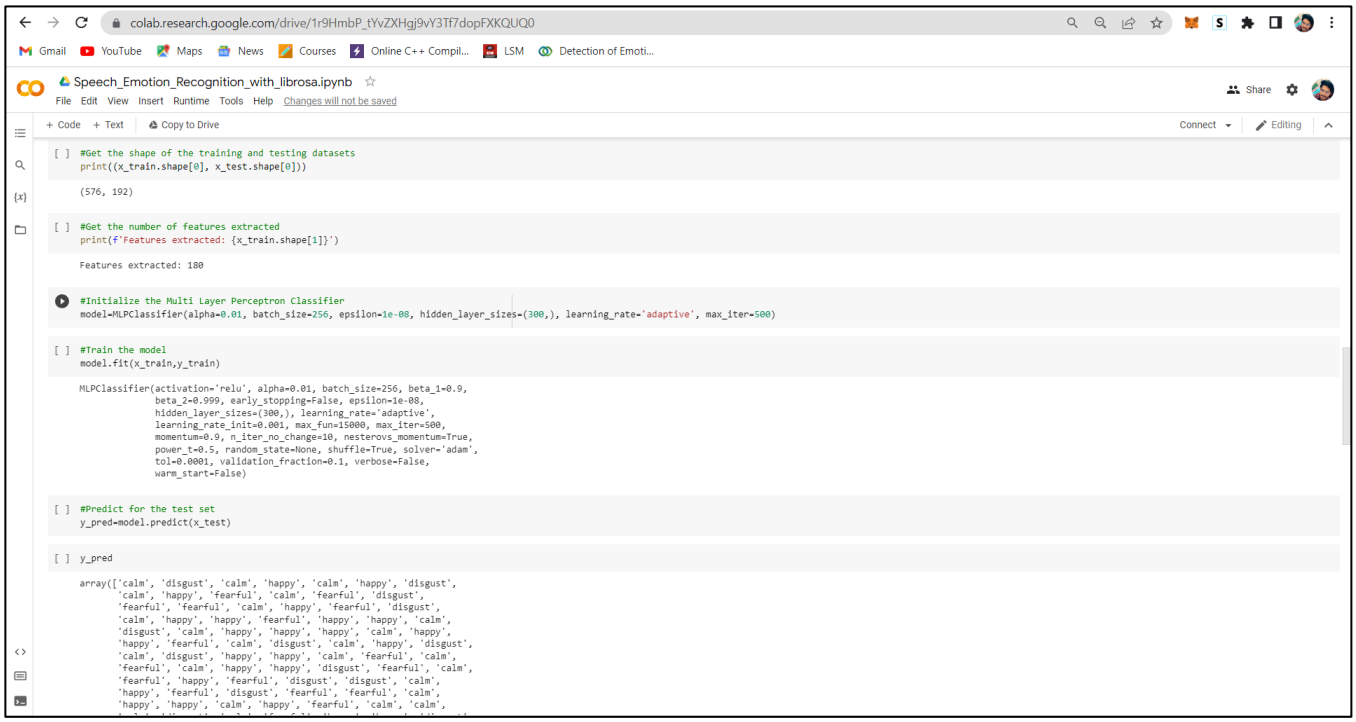
- *Portability*: The code is portable enough as it will work on various platform given all the necessary dependencies are installed.

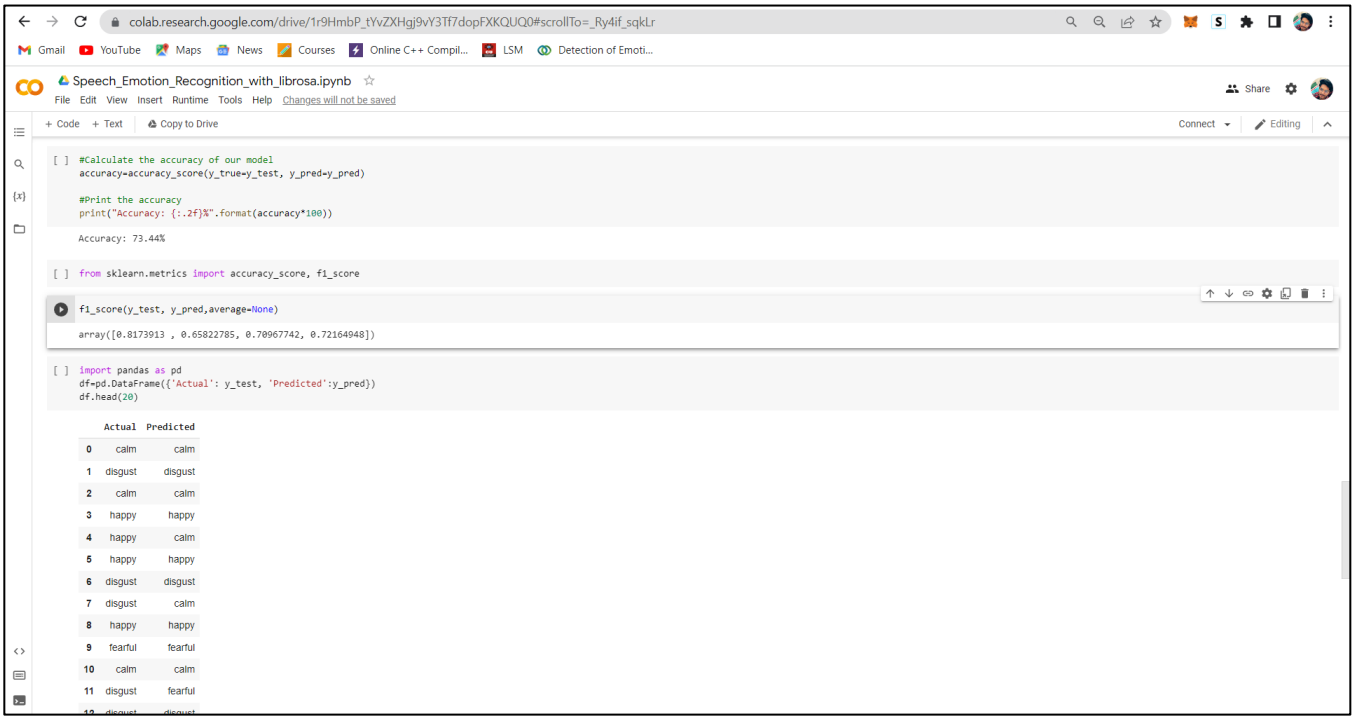
- *Debug Easily*: The coding should be error-free as much as possible.

4.2 Source Code

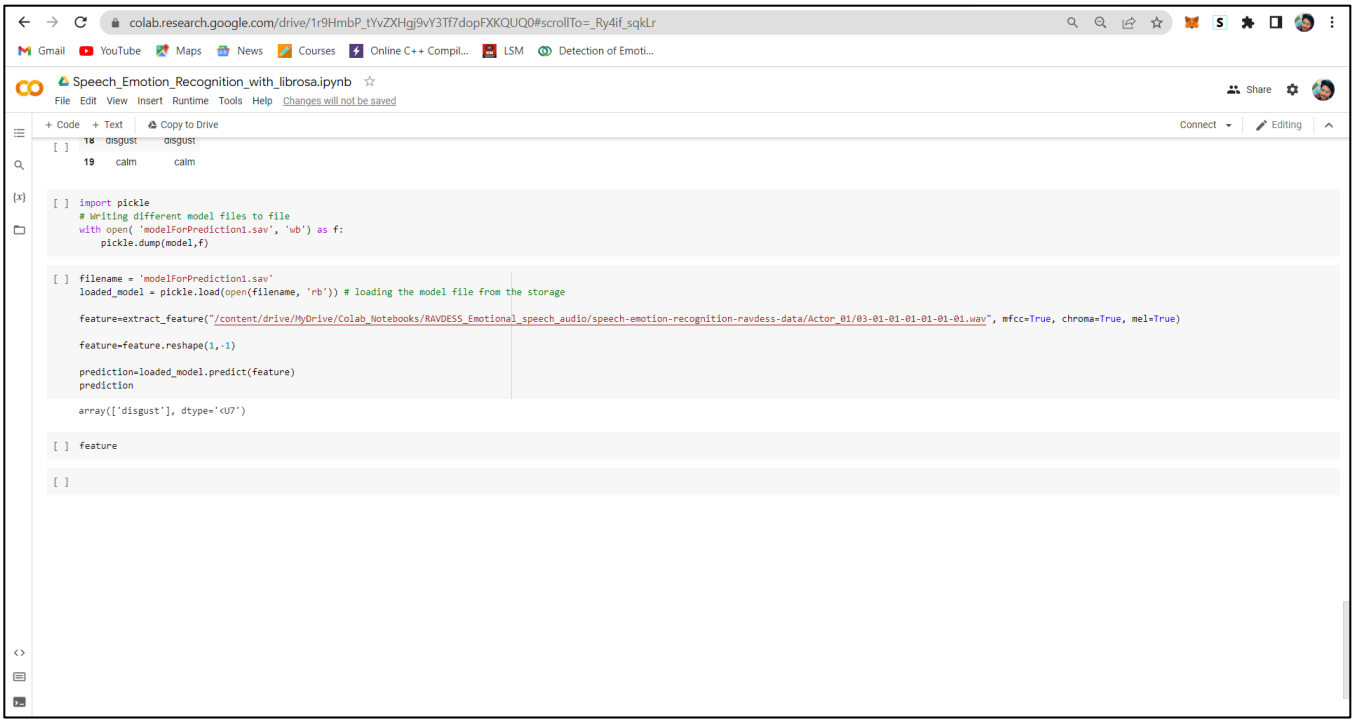

colab.research.google.com/drive/1r9HmbP_tYvZXHgj9vY3Tf7dopFXKQUQ0
Speech_Emotion_Recognition_with_librosa.ipynb
Dataset
import os
Root = "/content/drive/MyDrive/Colab_Notebooks/RAVDESS_Emotional_speech_audio"
os.chdir(Root)
ls
modelForPrediction1.sav
modelForPrediction.sav
speech-emotion-recognition-ravdess-data/
Speech_Emotion_Recognition_with_librosa.ipynb
standardScalar.sav
import librosa
import soundfile
import os, glob, pickle
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
#Extract features (mfcc, chroma, mel) from a sound file
def extract_feature(file_name, mfcc, chroma, mel):
with soundfile.SoundFile(file_name) as sound_file:
X = sound_file.read(dtype="float32")
sample_rate=sound_file.samplerate
if chroma:
stft=np.abs(librosa.stft(X))
result=np.array([])
if mfcc:
mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)
result=np.hstack((result, mfccs))
if chroma:
chroma=np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=0)
result=np.hstack((result, chroma))
if mel:
mel=np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
result=np.hstack((result, mel))


colab.research.google.com/drive/1r9HmbP_tYvZXHgj9vY3Tf7dopFXKQUQ0
Speech_Emotion_Recognition_with_librosa.ipynb
mel=np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
result=np.hstack((result, mel))
return result
# Emotions in the RAVDESS dataset
emotions={
'01':'neutral',
'02':'calm',
'03':'happy',
'04':'sad',
'05':'angry',
'06':'fearful',
'07':'disgust',
'08':'surprised'
}
#Emotions to observe
observed_emotions=['calm', 'happy', 'fearful', 'disgust']
#Load the data and extract features for each sound file
def load_data(test_size=0.2):
x,y=[],[]
for file in glob.glob("/content/drive/MyDrive/Colab_Notebooks/RAVDESS_Emotional_speech_audio/speech-emotion-recognition-ravdess-data/Actor_*/*.wav"):
file_name=os.path.basename(file)
emotion=emotions[file_name.split("-")[2]]
if emotion not in observed_emotions:
continue
feature=extract_feature(file, mfcc=True, chroma=True, mel=True)
x.append(feature)
y.append(emotion)
return train_test_split(np.array(x), y, test_size=test_size, random_state=9)
#Split the dataset
x_train,x_test,y_train,y_test=load_data(test_size=0.25)
x_train
array([[-6.02389954e+02, 5.97717743e+01, 8.60734844e+00, ...,
2.24425294e-05, 7.05290176e-06, 3.74911019e-06],
[-6.64690369e+02, 6.82226181e+01, 6.91438007e+00, ...,





	Actual	Predicted
0	calm	calm
1	disgust	disgust
2	calm	calm
3	happy	happy
4	happy	calm
5	happy	happy
6	disgust	disgust
7	disgust	calm
8	happy	happy
9	fearful	fearful
10	calm	calm
11	disgust	fearful


colab.research.google.com/drive/1r9HmbP_tYvZXHgj9vY3Tf7dopFXKQUQ0#scrollTo=_Ry4if_sqkLr
Gmail YouTube Maps News Courses Online C++ Compil... LSM Detection of Emoti...
Speech_Emotion_Recognition_with_librosa.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
Share
+ Code + Text Copy to Drive
Connect Editing
18 disgust disgust
19 calm calm
import pickle
# Writing different model files to file
with open('modelForPrediction1.sav', 'wb') as f:
pickle.dump(model,f)
filename = 'modelForPrediction1.sav'
loaded_model = pickle.load(open(filename, 'rb')) # loading the model file from the storage
feature=extract_feature("/content/drive/MyDrive/Colab_Notebooks/RAVDESS_Emotional_speech_audio/speech-emotion-recognition-ravdess-data/Actor_01/03-01-01-01-01-01-01.wav", mfcc=True, chroma=True, mel=True)
feature=feature.reshape(1,-1)
prediction=loaded_model.predict(feature)
prediction
array(['disgust'], dtype='<U7')
feature

5. CONCLUSION

In this project we have tried to analyze some samples of speech using the deep learning technique. Firstly we loaded the datasets then we visualized the different human emotions using our functions waveshow and spectrogram using the Librosa library. Then we extracted the acoustic features of all our samples using the MFCC method and arranged the sequential data obtained in the 3D array form as accepted by the LSTM model. Then we build the LSTM model and after training the model we visualized the data into the graphical form using matplotlib library and after some repeated testing using different values the average accuracy of the model is found to be 73%. Enhancement of the robustness of emotion recognition system is still possible by combining databases and by fusion of classifiers. The effect of training multiple emotion detectors can be investigated by fusing these into a single detection system. We aim also to use other feature selection methods because the quality of the feature selection affects the emotion recognition rate: a good emotion feature selection method can select features reflecting emotion state quickly. The overall aim of our work is to develop a system that will be used in a pedagogical interaction in classrooms, in order to help the teacher to orchestrate his class. For achieving this goal, we aim to test the system proposed in this work.

6. REFERENCES

1. H. Cao, R. Verma, and A. Nenkova, “Speaker-sensitive emotion recognition via ranking: Studies on acted and spontaneous speech,” Comput. Speech Lang., vol. 28, no. 1, pp. 186– 202, Jan. 2015.

2. L. Chen, X. Mao, Y. Xue, and L. L. Cheng, “Speech emotion recognition: Features and classification models,” Digit. Signal Process., vol. 22, no. 6, pp. 1154–1160, Dec. 2012

3. T. L. Nwe, S. W. Foo, and L. C. De Silva, “Speech emotion recognition using hidden Markov models,” Speech Commun., vol. 41, no. 4, pp. 603–623, Nov. 2003.

4. S. Wu, T. H. Falk, and W.-Y. Chan, “Automatic speech emotion recognition using modulation spectral features,” Speech Commun., vol. 53, no. 5, pp. 768–785, May 2011.

5. J. Rong, G. Li, and Y.-P. P. Chen, “Acoustic feature selection for automatic emotion recognition from speech,” Inf. Process. Manag., vol. 45, no. 3, pp. 315–328, May 2009.

6. C.-H. Wu and W.-B. Liang, “Emotion Recognition of Affective Speech Based on Multiple Classifiers Using Acoustic-Prosodic Information and Semantic Labels,” IEEE Trans. Affect. Comput., vol. 2, no. 1, pp. 10–21, Jan. 2011.

7. S. S. Narayanan, “Toward detecting emotions in spoken dialogs,” IEEE Trans. Speech Audio Process., vol. 13, no. 2, pp. 293–303, Mar. 2005.

8. B. Yang and M. Lugger, “Emotion recognition from speech signals using new harmony features,” Signal Processing, vol. 90, no. 5, pp. 1415–1423, May 2010

9. E. M. Albornoz, D. H. Milone, and H. L. Rufiner, “Spoken emotion recognition using hierarchical classifiers,” Comput. Speech Lang., vol. 25, no. 3, pp. 556–570, Jul. 2011.

10. C.-C. Lee, E. Mower, C. Busso, S. Lee, and S. Narayanan, “Emotion recognition using a hierarchical binary decision tree approach,” Speech Commun., vol. 53, no. 9–10, pp. 1162– 1171, Nov. 2011.