

A Project Report on

Emotion Recognition Of Speech Using Deep Learning

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

G. LAKSHMI MANASA (184G1A0533)

B. PUSHPALATHA (184G1A0560)

T. MADHAVI (184G1A0538)

P. DILWAR (194G5A0502)

Under the Guidance of

Dr. T. Venkata NagaJayudu M.Tech, Ph.D.,

Associate Professor



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPURAMU

(Affiliated to JNTUA & Approved by AICTE)

(Accredited by NAAC with 'A' Grade & Accredited by NBA (EEE, ECE & CSE))

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu-515701.

2021-2022

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPURAMU

(Affiliated to JNTUA & Approved by AICTE)

(Accredited by NAAC with 'A' Grade & Accredited by NBA (EEE, ECE & CSE))

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu-515701.



This is to certify that the project report entitled **Emotion Recognition Of Speech Using Deep Learnig** is the bonafide work carried out by **G. Lakshmi Manasa** bearing Roll Number **184G1A0533**, **B.Pushpalatha** bearing Roll Number **184G1A0560**, **T. Madhavi** bearing Roll Number **184G1A0538** and **P.Dilwar** bearing Roll Number **194G5A0502** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2021-2022.

Guide

Dr. T. Venkata Naga Jayudu M.Tech.,Ph.D.,
Associate Professor

Head of the Department

Mr. P. Veera PrakashM.Tech,(Ph.D).,
Assistant Professor & HOD

Date:

Place: Ananthapuramu

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we now have the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express my indebted gratitude to my Guide **Dr. T. Venkata Naga Jayudu**, M.Tech, Ph.D., **Associate Professor, Computer Science & Engineering**, who has guided me a lot and encouraged me in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep-felt gratitude to **Mr. K. Venkatesh**, M.Tech., **Assistant Professor**, project coordinator valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We are very much thankful to **Mr. P. Veera Prakash**, M.Tech., (Ph.D), **Assistant Professor & Head of the Department, Computer Science & Engineering**, for his kind support and for providing necessary facilities to carry out the work.

We wish to convey my special thanks to **Dr. G. Bala Krishna**, Ph.D., **Principal of Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities. Finally, we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates

Declaration

We, Ms G. Lakshmi Manasa with reg no: 184G1A0533, Ms B. Pushpalatha with reg no: 184G1A0560, Ms T. Madhavi with reg no: 184G1A0538, Mr P. Dilwar with reg no: 194G5A0502 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that the dissertation entitled “EMOTION SPEECH RECOGNITION USING DEEP LEARNING” embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of Dr. T. Venkata Naga Jayudu M.Tech,Ph.D., Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University or Institute for the award of any Degree or Diploma.

G. LAKSHMI MANASA

Reg no: 184G1A0533

B. PUSHPALATHA

Reg no: 184G1A0560

T. MADHAVI

Reg no: 184G1A0538

P. DIWAR

Reg no: 194G5A0502

CONTENTS

List of figures	viii
List of abbreviations	ix
Abstract	x
Chapter 1: Introduction	1
1.1 : Introduction to Deep Learning	2
1.2 : Introduction to Convolutional Neural Network	2
1.2.1 : Convolutional Layer	3
1.2.2 : Pooling Layer	3
1.2.3 : Fully connected Layer	4
1.2.4 : CNN Algorithm	4
1.3 : Objective	5
1.4 : Problem Statement	5
1.5 : Organization of the Project	6
Chapter 2 : Literature Survey	7
2.1 : Existing System	7
2.2 : Proposed System	10
Chapter 3 : Analysis	11
3.1 : Introduction	11
3.2 : Feasibility Study	11
3.3 : Software Requirements Specification	13
3.4 : Hardware Requirement	13
3.5 : Software Requirements	13
3.5.1 : Google Collaboratory	14
3.5.2 : Languages Used	16
Chapter 4 : Design	18
4.1 : UML Introduction	18
4.1.1 : Usage of UML in Project	18
4.2 : Dataflow Diagram	18
4.3 : Architecture	20
4.4 : Steps involved in Design	20
Chapter 5 : Implementation	21
5.1 : Libraries Used	21
5.2 : Implementation	22
5.2.1 : Data Preprocessing	22
5.2.2 : Feature Extraction	25
5.2.3 : Dataset Splitting	27
5.2.4 : Model Creation and Prediction	28

Chapter 6: Result	33
Conclusion	34
References	35

LIST OF FIGURES

Fig. No	Title	Page No.
Fig 3.1	Pop up tab for creating new notebook	14
Fig 3.2	New Notebook	14
Fig 3.3	Running Environment	15
Fig 3.4	Uploading file into google colab	15
Fig 4.1	Data Flow Diagram	19
Fig 4.2	Architecture	19
Fig 5.1	Preprocessing	24
Fig 5.2	Feature Extraction Technique	25
Fig 5.3	Data Splitting	28
Fig 5.4	Prediction Pipeline	29
Fig 5.5	Output Of CNN	31
Fig 6.1	Testing Accuracy	34
Fig 6.2	Accuracy	35
Fig 6.3	Confusion Matrix	35

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
CV	Computer Vision
SRS	Software Requirement Specification
UML	Unified modeling language
Numpy	Numerical Python
ML	Machine Learning
SKimage	Scikit-image
HMM	Hidden Markov Model
ANN	Artificial Neural Network
MFCC	Mel-Frequency Cepstral Coefficient
MLP	Multilayer Perceptron Classifier
RAVDESS	Ryerson Audio-Visual Database of Emotional speech and Song dataset
SVM	Support Vector Machine
GMM	Gaussian Mixture Model

ABSTRACT

Emotion Recognition of Speech using Deep learning is the act of attempting to recognize human emotion and affective states from speech. This is capitalizing on the fact that voice often reflects underlying emotion through tone and pitch. This is also the phenomenon that animals like dogs and horses employ to be able to understand human emotion. Existing system of Emotion Recognition Of Speech was developed with Multilayer Perceptron(MLP). Generally MLP is best, when the model is quality trained. this is not all ways possible in real world scenario. The proposed system of Emotion Recognition Of Speech will be developed with Ensemble Methods i.e; Convolutional Neural Network(CNN) and so on, which are much better than MLP. It can detect important features with out human supervision, computationally efficient, easy to integrate, high performance.

Keywords: *Convolutional Neural Network(CNN), librosa, panda, numpy , sklearn, Machine Learning.*

CHAPTER 1

INTRODUCTION

Detecting emotions is one of the most important marketing strategies in today's world. Anyone can personalize different things for an individual specifically to suit their interest. It is the project where anyone could detect a person's emotions just by their voice which will let us manage many AI-related applications. Speech emotion recognition research involves the traditional speech signal processing, pattern recognition, human psychology, artificial intelligence, human-computer interaction and other fields. In human-computer interaction, speech emotion recognition is an important part to determining the emotional state of interaction objects. Speech recognition is the process of converting an acoustic signal, captured by microphone or a telephone, to a set of characters. They can also serve as the input to further linguistic processing to achieve speech understanding, a subject covered in section. As we know, speech recognition performs tasks that similar with human brain.

Recognition system is generally composed of three parts, the first being speech signal acquisition, then comes the feature extraction followed by emotion recognition. Some examples could be including call centers to play music when one is angry on the call. Another could be a smart car slowing down when one is angry or fearful. As a result, this type of application has much potential in the world that would benefit companies and even safety to consumers. Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and determine the emotions of the speaker such as normal, anger, happiness and sadness. The founder of modern philosophy "René Descartes", identified six simple and primitive emotions: wonder, love, hatred, desire, joy, and sadness. Other philosophers identified categories of emotions which include composed of some of these six or species of them. As a result of the experiences and observations experienced by man over the centuries it became easy for him to distinguish emotions, such as when a person is angry, his tone raises, and his expression becomes stern.

At the same time when a person is happy, he speaks in a musical tone thus there is a look of glee on his face and the content of his speech is rather pleasant. Based on these observations, a person can quickly identify the state of the speaker whether he is happy, sad, angry or others states. In this work, the preprocessing is applying to

get a pure signal which is used to extract the features. After the features of this signal are extracted then it is used to distinguish the emotion. There are several methods for feature extraction of voice signal such as Liner Predictive Coding (LPC), Hidden Markov Model (HMM), Artificial Neural Network (ANN), and Mel-Frequency Cepstral Coefficient (MFCC) .

1.1 Introduction to Deep Learning

Deep Learning in a single term we can understand as Human Nervous System. Machine Vision Deep learning sets are made to learn over a collection of audio/image also known as training data, in order to rectify a problem. The various deep learning models trains a computer to visualize like a human. Deep learning models based on the inputs to the nodes can visualize. Hence network type is like that of a Human Nervous System, with every node performing under a larger network as a neuron. So, deep learning models are basically a part of Artificial Neural Networks. Algorithms of Deep learning learns in depth about the input audio/image as it passes over every Neural Network Layer. Low-level Characteristics like edges are detected by learning given to the initial layers, and successive layers collaborate characteristics from prior layers in a more philosophical representation.

Deep learning is a modern machine learning technique that emerged to deal with big databases and complex systems. The advent of deep learning brought with it a wave of novel algorithms that diminished the need for “hand- crafted” features prior to classification . That is, deep learning models can learn low-level features from training data in their lower layers and build high- level representation in the upper layers based upon the proceeding layers. As a result, the deep learning models are able to extract the features automatically. Recently, a rapid growth has been observed in using deep learning models to classify speech emotions. The efficacy of deep learning models in speech emotion recognition has been examined during last years in various studies.

1.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) are one of the most popular deep learning models that have manifested remarkable success in the research areas such as object recognition , face recognition, handwriting recognition , speech recognition ,and natural language processing . The term convolution comes from

the fact that convolution—the mathematical operation—is employed in these networks. Generally, CNNs have three fundamental building blocks: the convolutional layer, the pooling layer, and the fully connected layer. CNN has their "neurons" arranged greater like the ones of the frontal lobe, the area responsible for processing visual stimuli in people and different animals. CNN uses a system such as a multi-layer perceptron that has been created for reducing the process requirement. CNN has main three layers. An input layer, an output layer, and a hidden layer. They include other sub-layers: multiple convolutional layers, pooling layers, fully connected layers, normalization layers. These layers reduce the drawbacks of speech and increase the efficiency of a result of the system that has the most effective way to train speech or audio voice input in raw pixel data and trains this model then extract with features like MFCC and MFCC automatically apply the classifier MLP.

1.2.1 Convolutional layer

Convolutional layers in CNNs use convolution instead of multiplication to compute the output. As a result, the neurons in the convolutional layers are not connected to all the neurons in their preceding layers. This architecture is inspired by the fact that neurons of the visual cortex have local receptive field. That is, the neurons are specialized to respond to the stimuli limited to a specific location and structure. As a result, using convolution introduces sparse connectivity and parameter sharing to CNNs, which decreases the number of parameters in deep neural networks drastically.

1.2.2 Pooling Layer

The second important building block of CNNs is a pooling layer. This layer is used to make the outputs less sensitive to the local variation in the inputs.

This invariance to small local translation can decrease the spatial resolution and lead to underfitting in some applications. When accurate spatial features are not required, pooling can improve the performance of CNNs in extracting the features of interest. Further, pooling can reduce overfitting since it decreases the number of dimensions and parameters. In a sense, pooling takes subsamples from the outputs. Similar to convolutional layers, pooling layers use a kernel (a

rectangular receptive field) to apply an aggregation function such as maximum, average, L_2 -norm, or weighted average to summarize the values of the neurons within the pooling kernel. To have a pooling layer in CNNs, we need to determine the size of pooling kernels, the step of shifting, and the number of padding.

1.2.3 Fully Connected layer

A typical CNN consists of several convolutional layers where each convolutional layer is followed by a pooling layer. The last building block of CNNs is the fully connected layer, which is basically a traditional MLP. This component is used to either make a more abstract representation of the inputs by further processing of the features or classify the inputs based on the features extracted.

1.2.4 CNN Algorithm

Here, we use Convolutional Neural Network (CNN) algorithm for our system speech emotion recognition. Which is used in many modules for recognizing the emotions and classifiers are used for classifying the emotions such as happy, sad, angry, surprise, disgust, neutral. We used Python for implementing the CNN algorithm, here python used various libraries to processing our model. These Libraries are Sound file, Librosa, Numpy, PyAudio, Scikit-learn to extracting audio features and training the model also testing how good our model is doing.

Steps

Step 1: First we take the audio speech file through the web application in the form of rating in audio format.

Step 2: This file is plotted waveforms and spectrograms.

Step 3: Then we use the Sound file Library for the read given audio file and then also use the LIBROSA a Python library, we extract the MFCC Mel frequency cepstral Coefficient along with the 10-20.

Step 4: This data processing, we divide this data in train and test after using CNN algorithm and then performing the operations.

Step 5: We check the emotions from the trained data of the human voice.

Step 6: After training the audio voice file, we test the voice file for recognizing the emotions.

Step 7: When the emotion is recognized our system predicts the emotion. Step 8: Display the predicted emotions with rating

1.3 Objective

The Main objective of the project is to build Speech Emotion Recognition System using Deep Learning. Speech Emotion Recognition, abbreviated as SER, is the act of attempting to recognize human emotion and affective states from speech. This is capitalizing on the fact that voice often reflects underlying emotion through tone and pitch. This is also the phenomenon that animals like dogs and horses employ to be able to understand human emotion. SER is tough because emotions are subjective and annotating audio is challenging.

As human beings speech is amongst the most natural way to express ourselves. We depend so much on it that we recognize its importance when resorting to other communication forms like emails and text messages where we often use emojis to express the emotions associated with the messages. As emotions play a vital role in communication, the detection and analysis of the same is of vital importance in today's digital world of remote communication. Emotion detection is a challenging task, because emotions are subjective. There is no common consensus on how to measure or categorize them. We define a SER system as a collection of methodologies that process and classify speech signals to detect emotions embedded in them. Such a system can find use in a wide variety of application areas like interactive voice based-assistant or caller-agent conversation analysis. In this study we attempt to detect underlying emotions in rec So we want get best results out of speech orded speech by analyzing the acoustic features of the audio data of recordings.

1.4 Problem Statement

Emotion Recognition Of Speech is becoming increasingly important in various applications. At present, Emotion Recognition Of Speech is an emerging crossing field of artificial intelligence and artificial psychology; besides it is a popular

research topic of signal processing and pattern recognition. So, accuracy is most important when speaking about Emotion Recognition Of Speech. Our project aims at increasing the accuracy of recognizing emotion through speech.

1.5 Organization of the ProjectReport

The rest of the Project is organized into 6 chapters

Chapter 2 discusses some relevant literature in Speech Emotion Recognition System. This chapter presents the existing system and proposed system of Speech Emotion Recognition.

Chapter 3 discusses about the Requirements that are needed before doing the project. This chapter presents Software and Hardware requirements of the project.

Chapter 4 presents overall system design used for implementation and presents the results of the project.

Chapter 5 discusses the final implementation of the project and presents the results obtained in each step of project.

Chapter 6 presents the overall results obtained at the end of the project

CHAPTER 2

LITERATURE SURVEY

2.1 Existing System

The existing speech emotion recognition uses the Multilayer perception Classifier(MLP).Multilayer Perceptron Classifier (MLP Classifier) is used for the classification of emotions. RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song dataset) is the dataset used in this system.

Multi-layer Perceptron Classifier (MLP Classifier) relies on an underlying Neural Network to perform classification. MLP Classifier implements a Multi-Layer Perceptron (MLP) algorithm and trains the Neural Network using Backpropagation. The classifier identifies different categories in the datasets and classifies them into different emotions. Generally MLP is best, when the model is quality trained. There are some limitations in MLP too. The first limitation is that the MLPs always need fixed number of inputs to be provided for fixed number of outputs. The next limitation is that the network must be retrained when a new emotion is added to the system.

Emotions play an extremely important role in human mental life it is a medium of expression of one's perspective or his mental state to others. It is a channel of human psychological description of one's feelings. The basic phenomenon of emotion is something that every mind experiences and our paper make a specific hypothesis regarding the grounding of this phenomenon in the dynamics of intelligent systems. There are a few universal emotions-including Neutral, Anger, Surprise, Disgust, Fear, Happiness, and Sadness which any intelligent system with finite computational resources can be trained to identify or synthesize as required. In this paper, we present an approach to language-independent machine recognition of human emotion in speech [5]. The potential prosodic features are extracted from each utterance for the computational mapping between emotions and speech patterns. The selected

features are then used for training and testing a modular neural network. Classification result of neural network and K-nearest Neighbors classifiers are investigated for the purpose of comparative studies.[7]

Summary : Human emotions can be recognized from speech signals when facial expressions or biological signals are not available. In this work Emotions are recognized from speech signals using real time database. In this work we presented an approach to emotion recognition from speech signal. Our results indicate that the K-NN classifier average accuracy 91.71% forward feature selection while SVM classifier has accuracy of 76.57%.SVM classification for neutral and fear emotion are much better than K-NN. The future work will be to conduct comparative study of various classifier using different parameter selection method to improve performance accuracy

Speech emotion recognition research involves the traditional speech signal processing, pattern recognition, human psychology, artificial intelligence, human-computer interaction and other fields. In human-computer interaction, speech emotion recognition is an important part to determining the emotional state of interaction objects. The study of speech emotion recognition can be traced back to the early 1980s. Emotion classification can use the acoustic statistics. There are more and more studies of speech emotion recognition with the growing understanding of emotional intelligence. Schuller [1], [2] at Technical University of Munich were conducted a research on speech emotion recognition. A lot of studies on speech emotion reconducted by the Voice and Emotional Group at the University of Southern California and the Emotion Research Laboratory at Université de Genève. Technical University of Munich developed the open SMILE [3]-[5] speech feature extractor, which can automatically extract speech emotion features in batches, and carried out various classification methods of speech emotion recognition. Tsinghua University [6], CASIA [7], Zhejiang University and Southeast University also achieved outstanding results. Based on the CASIA Chinese Emotional Corpus, this paper

analyzed key technologies in the process of speech emotion recognition and the effect of feature reduction on the speech emotion recognition. We compared the emotional classification effect of speech recognition model based on SVM and ANN[6].

Summary : Based on the CASIA Chinese Emotional Corpus, the speech emotional features are analyzed and the statistical values of speech emotional features are extracted. Based on SVM and ANN respectively, two speech emotion classification models are constructed. In the two classification models, the effect of feature dimension reduction on accuracy is analyzed, and the two classification methods are compared. The experimental results show that the feature dimension reduction is helpful to improve the classification performance. In this paper, the performance of SVM in speech emotion recognition is slightly better than ANN.

In the future research, we need to expand the corpus to carry out research and analyze the distinction of speech emotion features and the effect of different feature reduction methods on speech emotion recognition to improve the accuracy of speech emotion recognition.

Speech emotion is an industry-based project which can be used in various places like in hospital by doctors or also therapist can use speech recognition model to determine a patient's emotion, in various call centers SER models are used to determine a customer's emotion while asking for their feedback or to understand them. Aside from them these models can be updated and also can be used along with artificial intelligence in creating virtual assistants that can recognize our emotions. Speech emotion recognition has many uses and in near future more advanced features would be created regarding this topic. But along with major advantages these topics can be pretty difficult to understand too and some of the concepts are challenging to understand like a speech recognition software to properly understand a human voice and accent also plays an important role as machine sometimes could not understand speech from every accent and in near future with more advances this problem could be solved or we can say that we could increase accuracy of these models[5]

Summary : From the given research we can conclude that speech representation is a tedious job but with the help of various plotting functions and various python libraries we could map our speech into graphical representation and with the help of extracting features we can differentiate between different voice models. Efficiency of the model could be increased if we accommodate various models. This gave us a better result which was 83.3% accurate. MLP and Gradient Boost were two classifiers which gave us the most accuracy.

2.2 Proposed System

The proposed system of Emotion Recognition of Speech will be developed using Convolutional Neural Network(CNN).This system is much better than MLP and overcome the limitations of MLP Here, the speech emotion recognition is based on the Convolutional

Neural Network (CNN) algorithm which uses different modules for the emotion recognition and the classifiers are used to differentiate emotions such as happiness, surprise, anger, neutral state, sadness, etc.. The dataset for the speech emotion recognition system is the speech samples and the characteristics are extracted from these speech samples using LIBROSA package. The classification performance is based on extracted characteristics. Finally we can determine the emotion of speech signal.

CHAPTER 3

ANALYSIS

3.1 Introduction

The Analysis Phase is where the project life cycle begins. This is the phase where you break down the deliverables in the high-level Project Charter into the more detailed business requirements. Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This formal process, which will be developed in more detail, consists of four basic steps

1. Elicitation – I ask questions, you talk, I listen
2. Validation – I analyze, I ask follow-up questions
3. Specification – I document, I ask follow-up questions
4. Verification – We all agree Most of the work in the Analysis Phase is performed by the role of analyst.

3.2 Feasibility Study

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

- To analyze whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

Types of Feasibility Study

The feasibility study mainly concentrates on below five mentioned areas. Among these Economic Feasibility Study is most important part of the feasibility analysis and Legal Feasibility Study is less considered feasibility analysis.

1. Technical Feasibility

In Technical Feasibility current resources both hardware software along with required technology are analyzed/assessed to develop project. This technical feasibility study gives report whether there exists correct required resources and technologies which will be used for project development. Along with this, feasibility study also analyzes technical skills and capabilities of technical team, existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology etc.

2. Operational Feasibility

In Operational Feasibility degree of providing service to requirements is analyzed along with how much easy product will be to operate and maintenance after deployment. Along with this other operational scopes are determining usability of product, Determining suggested solution by software development team is acceptable or not etc.

3. Economic Feasibility

In Economic Feasibility study cost and benefit of the project is analyzed. Means under this feasibility study a detail analysis is carried out what will be cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost and so on. After that it is analyzed whether project will be beneficial in terms of finance for organization or not.

4. Legal Feasibility

In Legal Feasibility study project is analyzed in legality point of view. This includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc. Overall it can be said that Legal Feasibility Study is study to know if proposed project conform legal and ethical

5. Schedule Feasibility

In Schedule Feasibility Study mainly timelines/deadlines is analyzed for proposed project which includes how many times teams will take to complete final project which has a great impact on the organization as purpose of project may fail if it can't be completed on time.

Feasibility Study Process

The below steps are carried out during entire feasibility analysis.

1. Information assessment
2. Information collection
3. Report writing
4. General information

3.1 Software Requirement Specification

SRS is a document created by system analyst after the requirements are collected. SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc,

The requirements received from client are written in natural language. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team

3.2 Hardware Requirement

Any Contemporary PC.

- Processor : i3/Intel Processor
- RAM : 8GB
- Hard Disk : 128GB

3.3 Software Requirements

- Operating System : Windows 10
- Tools : Google Collaboratory
- Dataset : RAVDESS
- Languages Used : Python

3.5.1 Google Collaboratory

Google Colab is the platform and a free Jupyter notebook environment provided by Google where we can build a Machine Learning Models using Python programming language.

Steps

1. Upload your data into Google drive. Here we have uploaded audio files which contains speeches of 24 people with variations in parameters.
2. To start working with Colab you first need to log in to your google account, then go to this link <https://colab.research.google.com>.
3. Opening your Jupyter Notebook: On opening the website you will see a pop-up containing following tab.

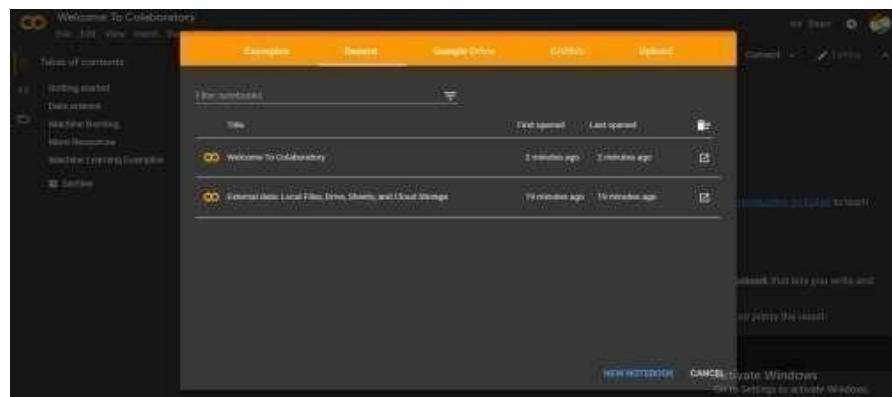


Fig 3.1 Pop up tab for creating new notebook.

4. Click on New Notebook at the bottom right corner to create new Notebook.

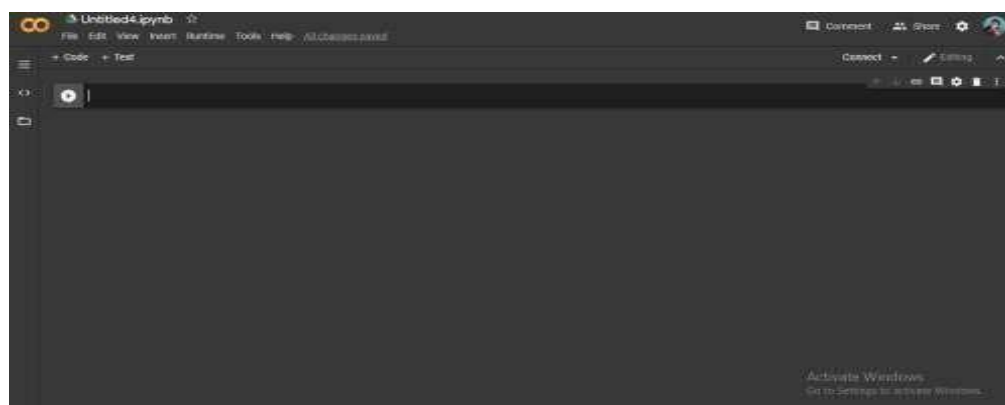


Fig 3.2 New notebook

On creating a new notebook, it will create a Jupyter notebook with Untitled0.ipynb and save it to your google drive in a folder named Colab Notebooks. Now as it is essentially a Jupyter notebook, all commands of Jupyter notebooks will work here. You can change the file name by file ->rename and save it.

5. Runtime Environment: Click the “Runtime” dropdown menu. Select “Change runtime type”. Select python2 or 3 from “Runtime type” dropdown menu.

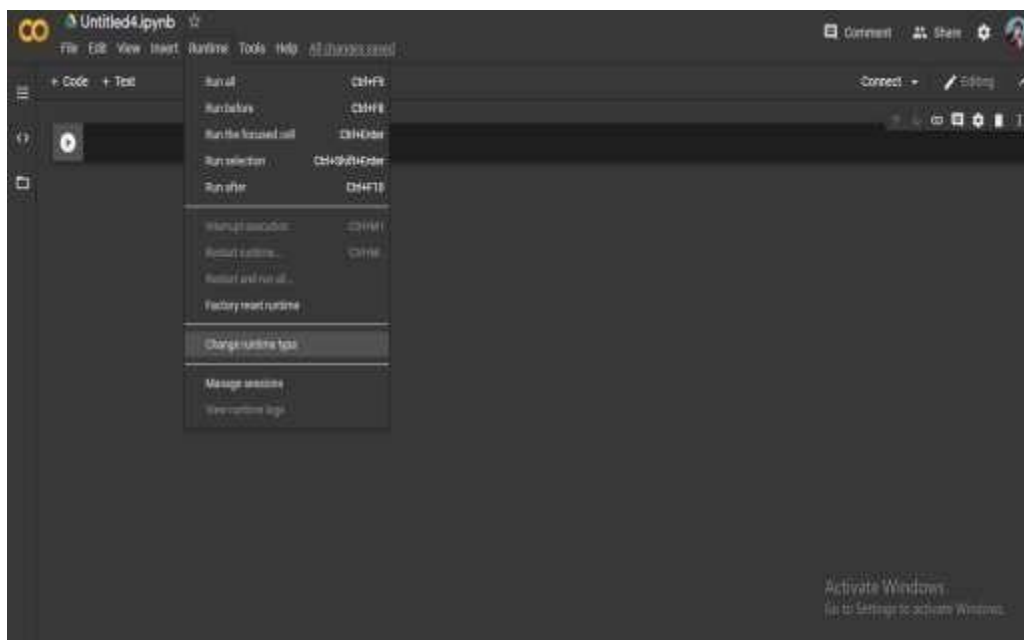


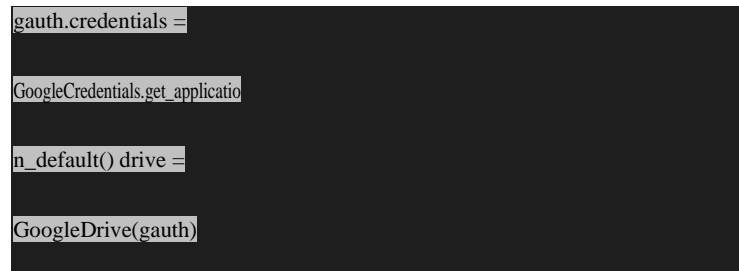
Fig 3.3 Running Environment

6. Upload file into google colab from google drive using following code.

```
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

auth.authenticate_user()

gauth = GoogleAuth()
```

```
gauth.credentials =  
GoogleCredentials.get_applicatio  
n_default() drive =  
GoogleDrive(gauth)
```

Fig: 3.5 Uploading file into google colab

3.5.2 Languages Used

The programming language that was used in our project is Python. The implementation of source code was done through python. Python is an interpreted, interactive, object-oriented programming language which is suitable for implementing machine learning algorithms in easier way.

Features of Python

Python provides lots of features that are listed below:

Easy to Learn and Use: Python is easy to learn and use. It is developer-friendly and high-level programming language.

Expressive Language: Python language is more expressive means that it is more understandable and readable.

Interpreted Language: Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

Cross-platform Language:

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

Free and Open Source:

Python language is freely available at official web address. The source-code is also available. Therefore, it is open source.

Object-Oriented Language:

Python supports object-oriented language and concepts of classes and objects come into existence.

Extensible:

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

Large Standard Library:

Python has a large and broad library and provides rich set of module and functions for rapid application development.

GUI Programming Support:

Graphical user interfaces can be developed using Python.

Integrated:

It can be easily integrated with languages like C, C++ and JAVA etc.

CHAPTER 4

DESIGN

4.1 Introduction to UML

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective.

UML is specifically constructed through two different domains, they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the systems.
- UML Design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

4.1.1 Usage of UML in Project

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

4.2 Data Flow Diagram

A data-flow diagram is a way of representing a flow of a data of a process or a system (usually an information system). This also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

The data-flow diagram is part of the structured-analysis modeling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram.

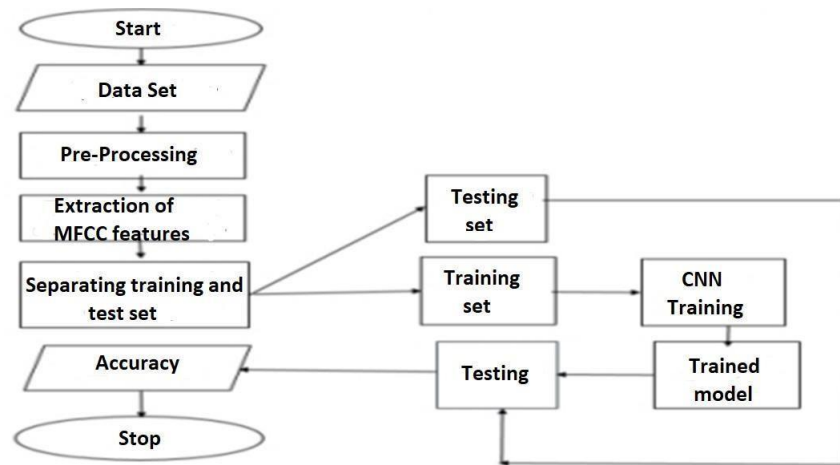


Fig 4.1 Data flow Diagram

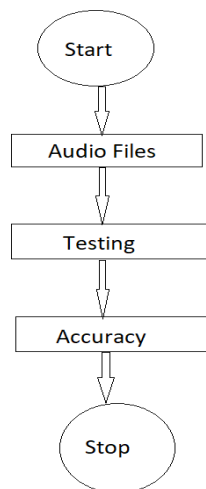


Fig 4.2 Testing Audio Files

4.3 Architecture

The following figure(4.1) depict the architecture of the speech Emotion recognition System.

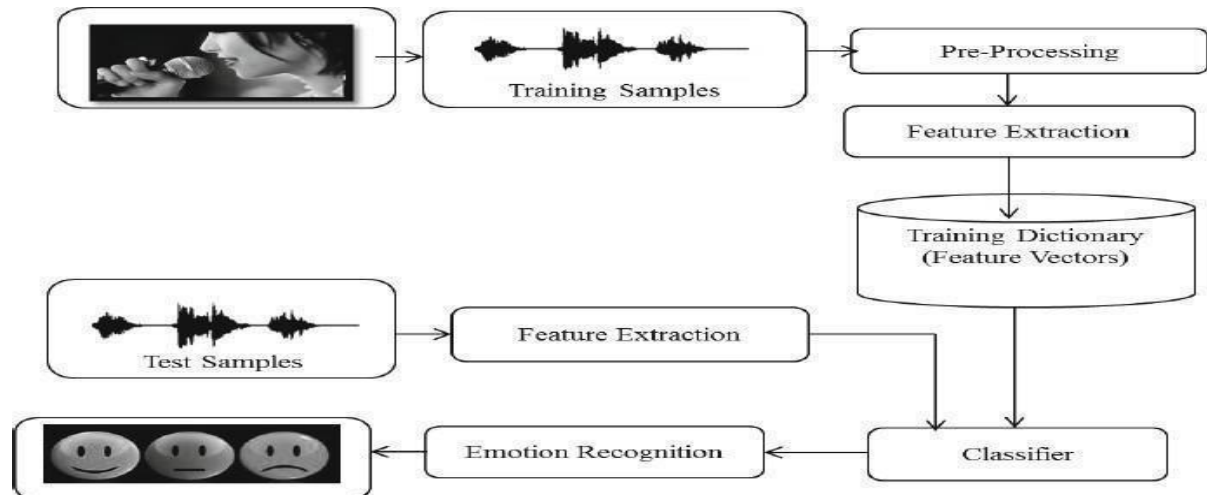


Fig 4.3 Architecture

4.3 Steps involved in Design

- Data Preprocessing
- Feature Extraction
- Dataset Splitting
- Model Creation and Prediction

Each step has its own specific reason and plays important role in building up a model of the project . Each step has been explained in implementation part.

CHAPTER 5

IMPLEMENTATION

5.1 Libraries Used

Python is increasingly being used as a scientific language. Matrix and vector manipulation are extremely important for scientific computations. Both NumPy and Pandas have emerged to be essential libraries for any scientific computation, including machine learning, in python due to their intuitive syntax and high- performance matrix computation capabilities.

NumPy

NumPy stands for ‘Numerical Python’ or ‘Numeric Python’. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since, arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem.

NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation. NumPy can be imported into the notebook using `import numpy as np`.

Pandas

Similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Pandas provides in-memory 2d table object called Data frame. It is like a spreadsheet with column names and row labels.

Hence, with 2d tables, pandas are capable of providing many additional functionalities like creating pivot tables, computing columns based

on other columns and plotting graphs. Pandas can be imported into Python using:

import pandas as pd.

Sklearn

Skikit-learn is a free software machine library for Python programming language. It features various classification , regression and clustering algorithms including support vector machine, random forest, k-means and gradient boosting. In our project we have used different features.

- **from sklearn.model_selection import train_test_split:**
Used for Splitting the dataset into Training and Testing

5.2 Implementation

5.2.1 Preprocessing

The first step involves organizing the audio files. The emotion in an audio sample can be determined by the unique identifier of the file name at the 3rd position, which represents the type of emotion. The dataset consists of five different emotions.

1. Happy 2. Sad 3. Angry 4. Fearful 5. Neutral 6. Disgust 7. surprised.

Defining Labels

Based on the number of classes to classify the speech labels are defined. Some of the classes are as follows:

Class: positive and negative Positive: Happy. Negative: Fearful, Sad, Angry.

Class: Angry, Sad, Happy, Fearful,.

Class: Angry, Sad, Happy, Fearful, , Neutral, Disgust, surprised.

As the typical output of the feature extracted were 2D in form, we decided to take bi-directional approach using both a 1D form of input and a 2D form of input as discussed below

1D Data Format

These features obtained from extraction from audio clips are in a matrix format. To model them on traditional ML algorithms like SVM and XGBoost or on 1-D CNN, we considered converting the matrices into the 1-D format by taking row means and column means. Upon preliminary modeling the results obtained from the array of row means turned out to be better than the array of column means, so we proceeded with the 1-D array obtained from row means of the feature matrices.

2D Data Format

The 2D features were used in the deep learning model (CNN). The y-axis of the feature matrices obtained depends on the `n_mfcc` or `n_mels` parameter we choose while extracting data. The x-axis depends upon the audio duration and the sampling rate we choose while feature extraction. Since the audio clips in our datasets were of varying lengths ranging from just under 2 seconds to over 6 seconds, steps like choosing one median length where we'll clip all audio files and pad all shorter files with zeroes to maintain dimensions wouldn't be feasible. This is because this would have resulted in the loss of information for longer clips and the shorter clips would be just silence for the latter half of their audio length. To check this problem, we decided to use different sampling rates in extraction in accordance with their audio lengths. In our approach any, audio file greater or equal to 5 seconds was clipped at 5 seconds and sampled at 16000 Hz and the shorter clips were sampled such that the $\text{audio duration} * \text{sampling rate}$ multiple remains 80000. In this way, we were able to maintain the dimensions of the matrix for all audio clips without losing much of the information. Before the extraction of the features of the signal, this signal is manipulate by using preprocessing. Preprocessing is mainly includes:

Silence removal: The speech signal usually include many parts of silence. The silence signal is not important because it is not contain information. There are several methods to remove these parts such as zero crossing rate (ZCR) and short time energy (STE). Zero- crossing rate is a measure of number of times in a given time interval such that the amplitude of the speech signals passes through a value

of zero. Short time energy is a measure of energy. Pre – processing has been an important step, the length of the audio signal needs to be consistent length, the discourse length of the audio signal has been maintained for each data by auditing the edges. Normalization has been performed along with missing value replacement. The different types of Emotion present in the SAVEE dataset are shown in figure 2:

Steps in Preprocessing

- 1.Data Cleaning
- 2.Data integration.
- 3.Data Transformation
- 4.Data reduction or Dimension Reduction

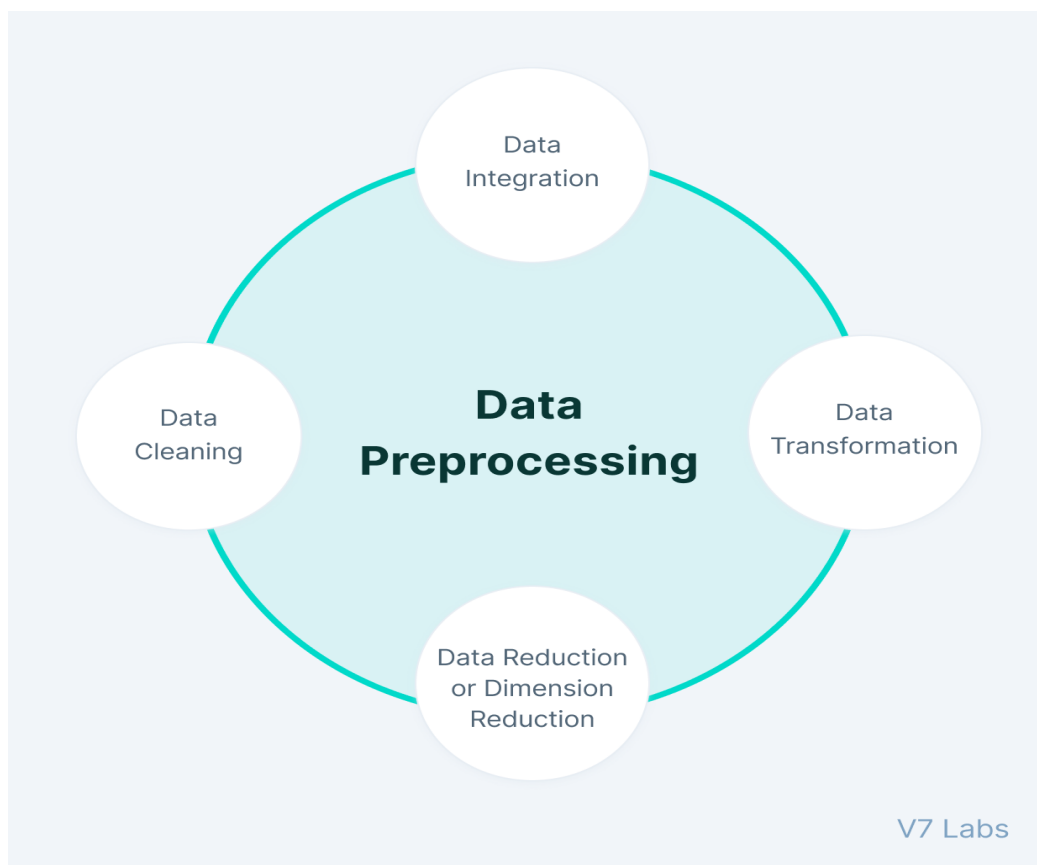


Fig 5.1 Preprocessing

The below code is implemented under preprocessing Module:

```
import pandas  
  
!pip install -U -q PyDrive
```

```
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive

from google.colab

import auth from oauth2client.client
import GoogleCredential

import os

import pandas as pd

import librosa

import glob

import numpy as np
```

5.2.2 Feature Extraction

The Shape of the Speech signal determines what sound comes out. If the shape is determined accurately, then the correct representation of the sound being generated is obtained. The job of Mel Frequency Cepstral Coefficients' (MFCC's) is to correctly represent it. MFCCs is used as input feature. Loading and converting audio data into MFCCs format is done by python package librosa.

Feature extraction extracts the features that help in analyzing the speaker. Detail process of Feature Extraction technique from speech to transformation of speech is presented in the figure.

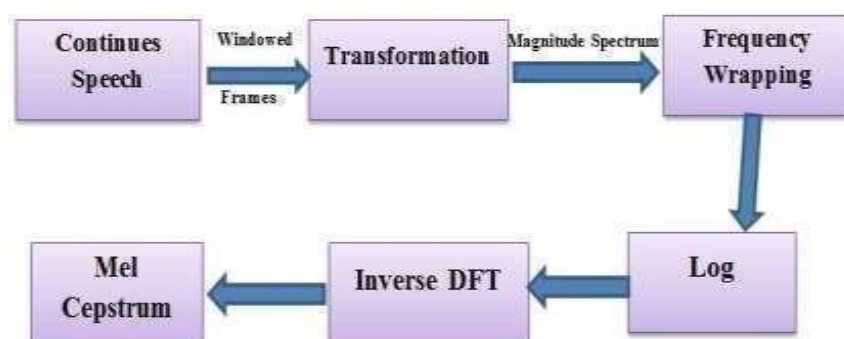


Fig.5.2 Feature Extraction Technique.

Fig 5.2 provide an overview of feature extraction process. In feature extraction process continuous speech is enter as input for windowing. Before entering in transformation stage windowing reduce the disruption process. After that speech signal which is in continues form converted in frames of the window. Then these frames are passed to Fourier transformation process which transforms frames of the window into a spectrum. After that spectrum is analyzed and Mel-spectrum is obtained at Mel- frequency scale with fixed resolution. Then it passes to log transformation and then to the inverse process of transformation that is inverse Discrete Fourier Transformation. Then the final result of Mel-spectrum is generated.

Different Feature extraction techniques with its advantages and disadvantages. ACCU-Accurate, ARBSF- Automatic Regression Based Speech Feature, ARTF-Artifacts, BPFT- Band Pass Filtering Technique, CFWLM- Capture Frequency With Low Modulation, CMCPS- Captures Main Characteristics Of Phones in Speech, DFRP- Different Filters Reduce Performance, DIALE- Difficulty in Analyze Local Events, ES- Enhance Speed, ESP- Easy Speech processing, FBNIDP- Filter Bandwidth No Independent Design Parameter, FET- Format Estimation technique, FEV-Fast Environment Variation, FLEXAM- Flexible Acoustic model, GCS-Good Computation Speed, GRE- Generates Residual Error, HMM-Hidden Markov Model, HPR- High Performance Ratio, LBRSE-Low Bit Rate Speech Encoding, LC-Low Complexity, LEGA-Less Effective Gaussian Assumption, LEGM-less Effective Generative Model, LOGHT-Log Higher Than, LPC- linear predictive coding, MFCC-Mel Frequency Capstrum Coefficient, MFS –Mel Frequency Scale, MHAS- Mimics Human Auditory System, MPD- Minor Performance Degradation, NCM- No Covariance Modeling, PLDA-Probabilistic Linear Discriminate Analysis, PLP- Perceptual linear prediction, RASTRA-Relative spectra, REL-Reliable, RESSDCVTIS-Residual Sound close to Vocal Tract Input Signals, RNI-Reduce Noise Impact, RSV- Remove Slow Variation, ROB-Robust, SDV- State Dependent Variable, SLFIT - Spacing of Linear Frequency Less Than, STATECH-Static Technique, VET- Vector Extraction Technique, WDWSV- Distinguishing Words With Similar

Vowel, WUNSS-Widely Used in Noisy Speech Signals.

Feature Extraction is an important step to make machines learn through feature categorization. In this paper the feature extraction has been done through MFCC. MFCC, short for Mel-Frequency Cepstral Coefficient. MFCC is a sentence, is an "image" of the vocal tract that delivers the sound. The initial phase in any programmed is speech acknowledgment framework is to remove the valuable component that recognize the pieces of the sound sign that are useful for distinguishing the etymological substance and disposing of the various stuff which conveys data like foundation commotion, feeling and so on. Mel Frequency Cepstral Coefficients (MFCCs) are an element broadly utilized in programmed discourse and speaker acknowledgment . Librosa function has been used to convert the audio signals into two tuples of array to generate features.

The mfcc feature can be extracted using python

library librosa.

Sample code:

```
sample_rate = np.array(sample_rate)

mfccs = librosa.feature.mfcc(y=X,
sr=sample_rate, n_mfcc=13)

files.append(file)

result = np.zeros((13,216))

result[:,mfccs.shape[0],:mfccs.shape[1]] =mfccs
```

5.2.3 Dataset Splitting

Splitting the dataset is the next step in data preprocessing in machine learning. Every dataset for Machine Learning model must be split into two separate sets – training set and test set.

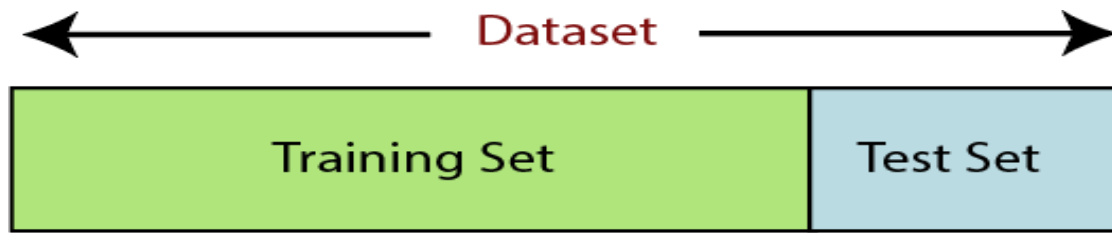


Fig.5.3 Dataset Splitting.

Training set denotes the subset of a dataset that is used for training the machine learning model. Here, you are already aware of the output. A test set, on the other hand, is the subset of the dataset that is used for testing the machine learning model. The ML model uses the test set to predict outcomes. Usually, the dataset is split into 70:30 ratio or 80:20 ratio. This means that you either take 70% or 80% of the data for training the model while leaving out the rest 30% or 20%. The splitting process varies according to the shape and size of the dataset in question.

5.2.4 Model Creation and Prediction

CNN Model Creation

CNN-1D (shallow)

This model consisted of 1 Convolution layer of 64 channels and same padding followed by a dense layer and the output layer.

CNN-2D (deep)

This model was constructed in a similar format as VGG-16, but the last 2 blocks of 3 convolution layers were removed to reduce complexity.

This CNN model had the following architectural complexity:

- 2 convolution layers of 64 channels, 3×3 kernel size and same padding followed by a max-pooling layer of size 2×2 and stride 2×2.
- 2 convolution layers of 128 channels, 3×3 kernel size and adding followed by a max-pooling layer of size 2×2 and stride 2×2.

- 3 convolution layers of 256 channels, 3×3 kernel size and padding followed by a max-pooling layer of size 2×2 and stride 2×2 .
- Each convolution layer had the 'relu' activation function.

After flattening, two dense layers of 512 units each were added dropout layers of 0.1 and 0.2 were added after each dense layer.

Finally, the output layer was added with a 'softmax' activation function. The result is based on the accuracy metrics in which there is a comparison between predicted values and the actual values. A confusion matrix is created which consists of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). From confusion metrics, we have calculated accuracy as follows: The model was trained on training data and tested on test data with different numbers of epochs starting from 50 to 100, 150 and 200. The accuracies were compared among all models viz. SVM, XGBoost and Convolution Neural Network (shallow and deep) for 1D features and 2D features.

CNN is a type of neural network model which allows us to extract higher representations for the image content. Unlike the classical image recognition where you define the image features yourself, CNN takes the image's raw pixel data, trains the model, then extracts the features automatically for better classification. A convolution sweeps the window through images then calculates its input and filter dot product pixel values. This allows convolution to emphasize the relevant features.

Essentially, these convolution layers promote **weight sharing** to examine pixels in kernels and develop visual context to classify images. Unlike Neural Network (NN) where the weights are independent, CNN's weights are attached to the neighboring pixels to extract features in every part of the image.

After each convolutional and max pooling operation, we can apply Rectified Linear Unit (ReLU). The ReLU function mimics our neuron activations on a "big enough stimulus" to introduce nonlinearity for values $x > 0$ and returns 0 if it does not meet the condition. This method has been effective to solve diminishing gradients. Weights that are very small will remain as 0 after the ReLU activation function.

A confusion matrix is created which consists of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). From confusion metrics, we have calculated accuracy as follows: The model was trained on training data and tested on test data with different numbers of epochs starting from 50 to 100, 150 and 200. The accuracies were compared among all models viz. SVM, XGBoost and Convolution Neural Network (shallow and deep) for 1D features and 2D features.

Prediction Pipeline

The final prediction pipeline is depicted schematically in Figure(Fig 5.4) below.

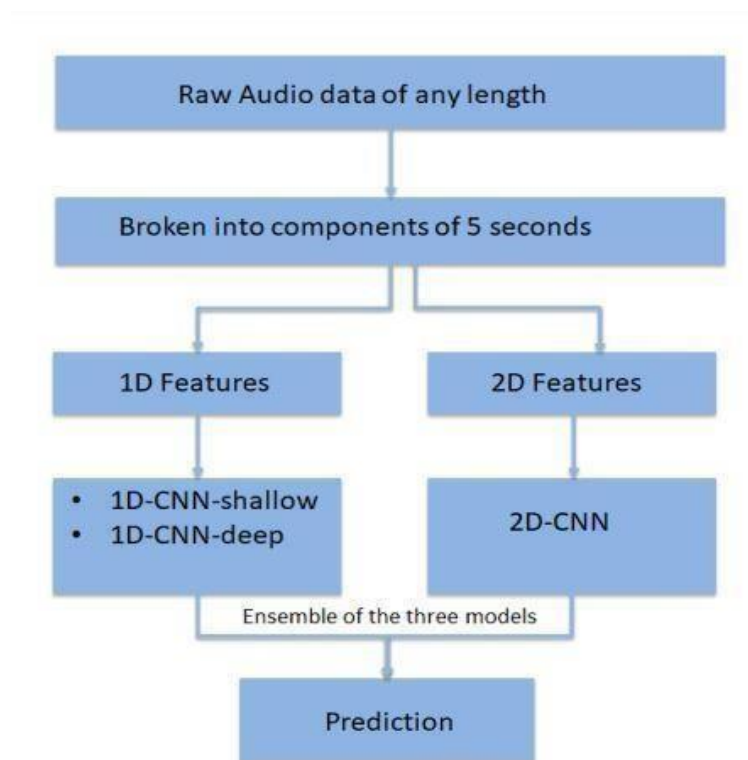


Fig.5.4 Prediction Pipeline.

Model Creation

The following code will create a CNN model:

```
import keras  
  
import numpy as np
```

```
import matplotlib.pyplot as plt

import tensorflow as tf

from keras.preprocessing import sequence

from keras.models import Sequential

from keras.layers import Dense, Embedding

from tensorflow.keras.utils import to_categorical

from keras.layers import Input, Flatten, Dropout, Activation,
BatchNormalization

from keras.layers import Conv2D, MaxPooling2D, LSTM, Lambda

from keras.models import Model

from keras.callbacks import ModelCheckpoint

from sklearn.model_selection import train_test_split

from keras import backend as K

from tensorflow.keras.utils import plot_model

model_ravdess = Sequential()

kernel = 5

model_ravdess.add(Conv2D(32, 5, strides=2, padding='same',
                        input_shape=(13, 216, 1)))

model_ravdess.add(Activation('relu'))

model_ravdess.add(BatchNormalization())

# model_ravdess.add(MaxPooling1D(pool_size=(8)))

model_ravdess.add(Conv2D(64, 5, strides=2, padding='same',))

model_ravdess.add(Activation('relu'))

model_ravdess.add(BatchNormalization())

model_ravdess.add(Conv2D(64, 5, strides=2, padding='same',))

model_ravdess.add(Activation('relu'))

model_ravdess.add(BatchNormalization())

# model_ravdess.add(MaxPooling2D(pool_size=(2, 3)))
```



```

# model_ravdess.add(Lambda(lambda x: K.squeeze(x, axis= 1)))

model_ravdess.add(Flatten())

# model_ravdess.add(LSTM(16))

# model_ravdess.add(Dropout(0.5))

model_ravdess.add(Dense(7))

model_ravdess.add(Activation('softmax'))

#opt = keras.optimizers.rmsprop(lr=0.00005, rho=0.9, epsilon=None,
decay=0.0)

model_ravdess.summary()

plot_model(model_ravdess, show_shapes=True,dpi = 65)

opt = tf.keras.optimizers.Adam(lr=0.001, beta_1=0.9, beta_2=0.999,
epsilon=None, decay=0.0, amsgrad=False)

```

Output Screenshot Of CNN Model

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 7, 108, 32)	832
activation (Activation)	(None, 7, 108, 32)	0
batch_normalization (Batch Normalization)	(None, 7, 108, 32)	128
conv2d_1 (Conv2D)	(None, 4, 54, 64)	51264
activation_1 (Activation)	(None, 4, 54, 64)	0
batch_normalization_1 (Batch Normalization)	(None, 4, 54, 64)	256
conv2d_2 (Conv2D)	(None, 2, 27, 64)	102464
activation_2 (Activation)	(None, 2, 27, 64)	0
batch_normalization_2 (Batch Normalization)	(None, 2, 27, 64)	256
flatten (Flatten)	(None, 3456)	0
dense (Dense)	(None, 7)	24199
activation_3 (Activation)	(None, 7)	0
Total params: 179,399		
Trainable params: 179,070		

Fig 5.5 Output of CNN

Prediction

The following code can be used to get the actual values and Predicted values of the dataset.

Code:

```
classes=['Neutral','Happy','Sad','Angry','Fearful','Disgust', 'Surprise']  
  
for i in range(len(classes_x)):  
    print(classes[classes_x[i]], ' ', classes[ravdess_valid_y[i]])
```

CHAPTER 6

RESULTS

- The results for various emotions are captured and tested to achieve more accuracy around 91%.we have performed this experiment based on proposed system.
- In this experiment we are using Convolutional neural networks(CNN) were trained over the RAVDESS dataset, and the performance is evaluated.
- The results are dependent on what emotions the user gives to the system and the system accepts in the form of voice then processed and then predicts by using CNN algorithm, MLP, Contract this features extraction is used to extract emotional characteristics from the emotion speech signal. Our proposed model achieves nearly 91% accuracy using CNN.
- The data was split into the ratio 80-20 and increasing the accuracy and also the efficiency of the classification of the process.

Sad	Sad
Neutral	Neutral
Neutral	Neutral
Happy	Happy
Neutral	Neutral
Happy	Happy
Neutral	Neutral
Fearful	Fearful
Sad	Sad
Happy	Happy
Sad	Sad
Angry	Angry
Neutral	Neutral
Surprise	Surprise
Angry	Angry
Happy	Happy
Angry	Angry
Surprise	Surprise
Disgust	Disgust
Sad	Sad
Disgust	Disgust
Angry	Angry
Neutral	Neutral
Fearful	Fearful
Surprise	Surprise
Sad	Sad
Angry	Angry
Angry	Angry
Surprise	Surprise
Happy	Happy
Fearful	Fearful

Fig 6.1 Testing Accuracy

Accuracy Output:

Fig 6.2 Accuracy

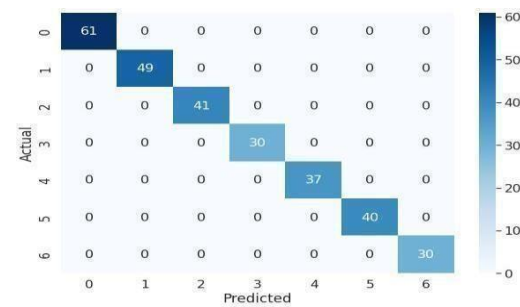
Confusion Matrix Representation:

Fig 6.3 Confusion Matrix

Conclusion

This Project, Detects Human's emotion while the speaker speaks and give an audio output. RAVDEESS dataset is used. RAVDESS dataset contains seven different emotions by all speakers. In our project Matplotlib module is used to plot the wave and saving it for future use—The results for various emotions are captured and tested to achieve more accuracy around 95%. performed this experiment based on proposed system. In this experiment used Convolutional neural networks(CNN) were trained over the RAVDESS dataset, and the performance is evaluated. The results are dependent on what emotions the user gives to the system and the system accepts in the form of voice then processed and then predicts by using CNN algorithm, MLP, Contract this features extraction is used to extract emotional characteristics from the emotion speech signal. Our proposed model achieves nearly 95% accuracy using CNN.

References

- [1] B. Schuller, G. Rigoll, M. Lang, "Hidden Markov model-based speech emotion recognition," in *Proc. 2003 IEEE International Conference on Acoustics, Speech, & Signal Processing*, 2003, pp. 401-404.
- [2] S. Björn and R. Gerhard, "Timing levels in segment-based speech emotion recognition," in *Proc. Inter speech 2006 and 9th International Conference on Spoken Language Processing*, 2006, vol. 4, no. 1818-1821.
- [3] F. Eyben, R. S. Klaus, W. S. Bjorn *et al.*, "The geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing," *IEEE Transactions on Affective Computing*, vol. 7, no. 2, pp. 190-202, 2016.
- [4] F. Eyben, G. L. Salomão, J. Sundberg *et al.*, "Emotion in the singing voice — a deeperlook at acoustic features in the light of automatic classification," *Eurasip Journal on Audio, Speech, and Music Processing*, vol. 19, 2015.
- [5] Sanjita. B. R, Nipunika. A, Rohita Desai, "Speech emotion recognitin "," Department of ECM Sreenidhi Institute of Technology and Science, Telangana, India, 2020
- [6] Xianxin Ke, Yujiao Zhu, Lei Wen and Wenzhen Zhang, "Speech Emotion Based on SVM and ANN", *International Journal of Machine learning and Computing*, Vol.8, No.3, June 2018.
- [7] Muzaffar Khan , Tirupathi Goskula, mohammed Nasiruddin, Ruhina Qyazi, "Comparison between K-nn and svm method for speech emotion recognition", (IJCSE), 2011.