A project report on

EMOTION RECOGNITION USING SPEECH PROCESSING

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

M.Nagalakshmi

(20U41A0541)

D.Saathvika (20U41A0554)

A.Mahitha Sri Varshini (20U41A0537)

B.Bharghavi (20U41A0544)

Under the Esteemed guidance of

Mr.P.Uday Bhaskar

Assistant Professor,

Department of Computer Science & Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DADI INSTITUTE OF ENGINEERING & TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU GV)

Accredited by NAAC with 'A' Grade and Inclusion u/s 2(f) & 12(B) of UGC Act

An ISO 9001:2015, ISO 14001:2015 & ISO 45001:2018 Certified Institute

NH-16, Anakapalle – 531002, Visakhapatnam, A.P.

(2020-2024)

DEPARTMENT OF

COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that project work entitled “Emotion recognition using speech processing” is being submitted by M.Nagalakshmi(20U41A0541),D.Saathvika(20U41A0554),A.MahithaSri Varshini(20U41A0537),B.Bhargavi(20U41A0544) .In partial fulfillment of the requirement for the award of the degree of BACHELOR OF TECHNOLOGY(A),Anakapalle affiliated to JNTUGV ,accredited by NAAC with ‘A’ grade is a record of bonafide work carried by them under my guidance and supervision .

PROJECT GUIDE **HEAD OF THE DEPARTEMENT**

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the Project entitled "**EMOTION RECOGNITION USING SPEECH PROCESSING**" has been carried out by us and contents have been presented in the form are for the award of the degree of BACHELOR OF TECHNOLOGY IN **COMPUTER SCIENCE AND ENGINEERING** under esteemed supervision **Mr.P.UdayBhaskar** ,Assistant Professor.This is a record of work carried out by us and results embodied in this project report have not been submitted to other university for any degree.

BY

M.Nagalakshmi	**20U41A0541**
D.Saathvika	**20U41A0554**
A.Mahitha Sri Varshini	**20U41A0537**
B.Bhargavi	**20U41A0544**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be complete without the mention of people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

First and foremost, we would like to thank our project guide **Mr.P.Uday Bhaskar**, Department of Computer Science and Engineering for giving us an opportunity to work on this challenging topic and providing us guidance. Her encouragement, support and suggestions are most valuable for the successful completion of our course.

We feel elated to extend our floral gratitude to Head of the department, **Dr.K.SUJATHA**, Department of Computer Science and Engineering for her encouragement all the way during analysis of the project. Her annotations, insinuations and criticisms are the key behind the successful completion of the project and for providing us all the required facilities.

We would like to take this opportunity to express our profound sense of gratitude to the revered Principal, **Dr.R.VAIKUNTA RAO** for giving us the opportunity of doing this project and for providing us all the required facilities.

We would like to express our deep sense of gratitude to the honorable Chairman, **SRI DADI RATNAKAR** for the resources and infrastructure provided for working on this project without any obstacles. The motivation and support given by the management is deeply adorable and we are fortunate to get a chance to work in this marvelous environment.

We also take this opportunity to express our heartfelt thanks to teaching and non teaching staff of the department, for their perspective comments and suggestions. We would like to thank our beloved parents for being patient, understanding and providing constant support. We would like to appreciate the critical comments given by our friends we have been working with. Our thanks to all others, who have directly or indirectly contributed in making our project a great success.

M.Nagalakshmi	**20U41A0541**
D.Saathvika	**20U41A0554**
A.Mahitha Sri Varshini	**20U41A0537**
B.Bhargavi	**20U41A0544**

ABSTRACT

Emotion recognition using speech processing has emerged as a pivotal research area with applications spanning from human-computer interaction to mental health diagnosis. This study investigates the extraction of emotional cues embedded within speech signals to discern underlying emotional states. Through sophisticated signal processing techniques and machine learning algorithms, such as Support Vector Machines (SVMs) and Convolutional Neural Networks (CNNs), the research aims to decipher nuanced emotional expressions including joy, sadness, anger, and more. Key aspects of the investigation include feature extraction methodologies to capture relevant acoustic properties, dataset curation for model training and evaluation, and algorithmic advancements to enhance recognition accuracy and robustness across diverse linguistic and cultural contexts. The findings of this research hold significant implications for fields such as affective computing, virtual assistants, and mental health monitoring systems. Ultimately, the pursuit of effective emotion recognition through speech processing promises to enrich human-computer interaction paradigms, paving the way for more empathetic and context-aware technological interfaces.

TABLE OF CONTENTS

LIST OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

In the intricate tapestry of human communication, emotions play a pivotal role, adding depth and nuance to our interactions. Emotion recognition, a fascinating field at the intersection of psychology and technology, seeks to unravel the complexities of human feelings. Among the various modalities employed for emotion recognition, speech processing stands out as a powerful and insightful tool. This technology enables the analysis of vocal cues, providing valuable insights into the emotional states of individuals.

Speech, as a primary mode of human expression, serves as a rich reservoir of emotional information. The cadence, pitch, tone, and rhythm of speech carry subtle cues that convey a spectrum of emotions, from joy and excitement to sadness and anger. Recognizing and deciphering these emotional nuances holds immense potential for applications in diverse domains, ranging from human-computer interaction and customer service to mental health diagnostics.

The human voice is a melodic symphony, capable of expressing a wide array of emotions. However, the challenge lies in harnessing the computational power to decode this emotional symphony accurately. Speech processing technologies, equipped with advanced machine learning algorithms, offer a promising avenue to unlock the emotional dimensions embedded in human speech.

One of the primary motivations behind emotion recognition using speech processing is to enhance the way we interact with technology. As we increasingly integrate artificial intelligence into our daily lives, creating systems that understand and respond to human emotions becomes crucial. Imagine a virtual assistant that not only understands your words but also recognizes the subtle changes in your voice, adapting its responses based on your emotional state. This level of emotional intelligence in machines has the potential to revolutionize human-machine interactions, making them more natural, empathetic, and responsive.

The applications extend to education and training, where understanding the emotional states of learners is crucial. The goal is to create personalized learning experiences by adapting instructional approaches based on the emotional responses detected in the learner's speech. For instance, if a student expresses frustration or confusion, the

system could provide additional support or alternative explanations. This application of emotion recognition seeks to optimize the learning process, making it more engaging and tailored to individual needs.

The exploration of emotion recognition using speech processing opens up a captivating frontier where technology and human emotions converge. As we strive to create more intuitive and responsive technologies, the ability to understand and respond to human emotions becomes paramount. Emotion recognition through speech processing stands as a beacon of innovation, offering the potential to redefine how we interact with machines, support mental health, and enrich our understanding of human emotion in the digital age. This journey into the realm of emotional intelligence heralds a new era where technology becomes not just intelligent but also emotionally aware.

In conclusion, Emotions can be of various types like happy, sad, angry, disguised etc. depending on the feeling and frame of mind of the person. In our study, we have used various datasets with different emotions. We have also combined four datasets to one dataset and then applied the model so that the efficiency of the model can be improved and there can be a variety in the data points. This has also resulted in eliminating the overfitting condition in our model. The Deep Neural Networks (DNN) classifiers provided a solution to circumvent the problem of feature selection. Using an end-to-end network, which accepts 2 raw data as input and outputs class labels, is the notion. There is no need to compute manually created features or determine the optimal parameters in terms of classification. The network itself does everything. To effectively partition the data into the desired categories, the network parameters are optimized. However, this very practical solution comes at the expense of a much higher requirement for labeled data samples than traditional classification methods.

CHAPTER 2

LITERATURE REVIEW

Md. Rayhan Ahmed et al.[1],used four deep neural network-based models built using LFABS. Model-A uses seven LFABs followed by FCN layers and a softmax layer for classification. Model-B uses LSTM and FCNs , Model-C uses GRU and FCNs and Model-D combines the three individual models by adjusting their weights. From each of these audio files, they hand-craft five categories of features- MFCC, LMS, ZCR, RMSE. did. data set. These features are used as inputs to a one-dimensional (1D) CNN architecture to further extract hidden local features in these speech files. To obtain additional contextual long-term representations of these learned local features via the 1D CNN block, we extended our experiment by incorporating LSTM and GRU after the CNN block , giving us more improved accuracy. After running DA, we observe that all four models perform very well on the SER task of detecting emotions from raw speech audio.Amongst all four models, the ensemble Model-D achieves the state-of-the-art weighted average accuracy of 99.46% for TESS dataset. Dr. Nilesh Shelke et al.[2] used RAVDESS, TESS and SAVEE datasets for classification.

Dr. Nilesh Shelke et al.[2] used RAVDESS, TESS and SAVEE datasets for classification. Their purpose is to mandate the modernization of current plans and technology enabling EDS and to implement assistance in all areas of computers and technology. Analytics complement emotions extracted from databases, layers, and model libraries created for emotion recognition from speech. It mainly focuses on data collection, feature extraction, and automatic emotion detection results. The intermodal recognition computer system is considered a unimodal solution because it offers higher sorting accuracy. Accuracy depends on the number of emotions detected, the features extracted, the classification method, and the stability of the database.

A data imbalance processing approach based on the selective interpolation synthetic minority oversampling (SISMOTE) methodology is suggested by Zhen-Tao Liu et al. [4] to reduce the influence of sample imbalance on emotion

identification outcomes. In order to minimize duplicate characteristics with inadequate emotional representation, a feature selection approach based on analysis of variance and gradient-enhanced decision trees (GBDT) is also provided. The results of speech emotion detection tests on the CASIA, Emo- DB, and SAVEE databases demonstrate that our technique produces an average of 90.28% (CASIA), 75.00% (SAVEE), and 85.82% (based on the findings) (Emo-DB). It demonstrates its precision in recognition. Utilizing voice emotion recognition is superior to some cutting-edge technologies.

To accomplish efficient speech emotion identification, Apeksha Aggarwal et al. [5] have presented two alternative feature extraction strategies. First, utilizing superconvergence to extract two sets of latent features from voice data, bidirectional feature extraction is presented. Principal Component Analysis (PCA) is used to produce the first set of features for the first set of features. A second method involves extracting the Mel spectrogram picture from the audio file and feeding the 2D image into his pre-trained VGG-16 model. In this study, several algorithms are used in comprehensive experimentation and rigorous comparative analysis of feature extraction approaches across datasets (RAVDESS AND TESS). The RAVDESS dataset offered significantly more accuracy.

Aseef Iqbal [6] used a real-time emotion detection system to analyze the tonal features of live-recorded speech and identify emotions. 34 audio characteristics, including MFCC, energy, spectral entropy, and others, are retrieved. To categorize emotions, this system essentially employs a model trained via gradient boosting. SVM and KNN, two of his other classifiers, are also used to assess their efficacy on test audio samples. The system is trained using two of his datasets, namely the RAVDESS and SAVEE databases. The method considers four emotions: neutrality, neutrality, pleasure, and neutrality.SVM and ANN exhibit 100% accuracy for both Anger and Neutral for RAVDESS (male). However, gradient boosting surpasses his SVM and ANN in both happiness and melancholy. SVM obtains her 100% accuracy in rage in RAVDESS (female), exactly like in the male version. Except for the sadness, SVM functions nicely

overall. At 87% and 100%, respectively, KNN also performs well on the furious and neutral dimensions. Anger and neutral don't mix well with gradient enhancing. When compared to other classifiers, ANN scores horribly on the happiness and sadness metric. SVM and ANN do extremely well on neutral and rage on the combined male and female data sets, but not on gradient boosting. KNN's performance with happiness and melancholy is quite lacking.

A voice analysis-based emotion recognition system was proposed by Noushin Hajarolasvadi and Hasan Demirel [7]. In order to extract an 88-dimensional vector of audio characteristics, including Mel-frequency cepstrum coefficients (MFCC), pitch, and intensity for each frame, they first partition each audio signal into overlapping frames of identical duration. For every frame, a spectrogram is created concurrently. The speech signal is retrieved from each audio signal by applying k-means clustering to the extracted characteristics of all the frames. This Is the last preprocessing step. Then, 3D tensor keyframe are used to represent the relevant series of spectrograms. Instead of using the entire set of spectrograms corresponding to speech frames, they selected the k best frames to represent the entire speech signal. They then compared the proposed 3D-CNN results s.with the 2D-CNN results and demonstrated that the proposed method outperforms pre-trained 2D meshes.

Using cluster-based genetic algorithms, Sofia Kanwal and Sohail Asghar [8] devised a feature optimization strategy. This method is based on three databases: the Ryerson Audio- Visual Database of Speech and Song (RAVDESS), the Berlin Emotional Speech Database (EMO-DB), and the Surrey Audio-Visual Comparing Datasets of Expressed Emotions (SAVE). The findings demonstrate that the suggested strategy significantly enhanced speech sentiment categorization. The recognition rates of EMO-DB are as follows: 89% for normal speakers, 86% for men, 88% for women, 82% for general speakers, 4% for men.

Key sequence segment selection based on repeated dial-based functional network (RBFN) similarity measurements inside a cluster is a new approach to SER that Muhammad Sajjad et al. [9] presented. The STFT technique is used to turn the chosen sequence into a spectrogram, which is then input to a CNN model to extract distinctive and significant characteristics from the spoken

spectrogram. In the suggested approach, we process important segments rather than entire utterances to lower the complexity of the model and normalize CNN features prior to actual processing to facilitate perception of spatiotemporal information. To increase detection accuracy and shorten model processing time, the proposed approach is tested on several standard datasets like IEMOCAP, EMO-DB, and RAVDESS. The suggested SER model's efficacy and robustness were tested against existing SER techniques with accuracy values of 72.25%, 85.57%, and 77.02% for up to using IEMOCAP, EMO-DB, or RAVDESS data sets .

The preparation of incoming audio samples using filters to denoise speech samples was the main topic of Anusha Koduru et alstudy .'s [10]. Discrete Wavelet Transform, Energy etc. methods are used to extract features in the next stage. In the feature selection stage, redundant data is culled from the features and sentiment is derived from the features using a global feature algorithm. It makes use of a machine learning classification method. These feature extraction algorithms have been tested for emotions that are common to all people, such as anger, happiness, sadness, and neutrality. The findings show that the decision tree has an accuracy of 85%, the SVM has an accuracy of 70%, and the LDA has an accuracy of 65%. The decision tree distinguishes the sentiment from the audio samples more precisely than SVM and LDA, according to the findings of the analysis.

K.A.Darshan, DR.B.N.Veerappa [11] the purpose of this paper is to document the development of speech recognition systems using CNNs. Design a model that can recognize the emotion of an audio sample. Various parameters are changed to improve the accuracy of the model. This paper also aims to find the factors that affect model accuracy and the key factors needed to improve model efficiency. The whitepaper concludes with a discussion of various CNN architectures and parameter accuracies needed to improve accuracy, as well as potential areas for improvement.

S.R. Harini Murugan [12] there is a disconnect between acoustic characteristics and human emotions, and because it mainly relies on discriminating acoustic features extracted for the detection job presented, automated speech emotion recognition is a difficult procedure. People express their emotions in a variety of

ways that are significantly distinct from one another. Looking at various things emphasizes pitch shifts, which are caused by the varying intensities of spoken emotions. Therefore, recognising speech emotions is a difficult problem for visual computation. As a result, vocal emotion detection is based on (CNN) algorithm, which employs several modules and uses classifiers to discriminate between emotions including happy, neutrality, and sorrow. The LIBROSA software is used to extract features from the voice samples that make up the data set for the Speech Emotion Recognition System. Performance in classification is based on retrieved characteristics. The audio signal's emotional content can then be identified.

According to Linqin Cai et al. [14], it is challenging for the existing emotion detection system to satisfy the need of emotion detection in one mode given the rapid expansion of social media. We suggested a multimodal emotion recognition model for voice and text in this research. We merged CNN and LSTM in the form of binary channels to learn acoustic-emotional properties by taking into account complementarity between various modes. In the meanwhile, we effectively captured text characteristics using a Bi-LSTM network (bidirectional long-short-term memory). In order to learn and identify the fusion characteristics of, we also used a deep neural network. The results of both voice sentiment analysis and textual sentiment analysis were used to establish the final emotional state. The concept put out in the IEMOCAP database was then validated using a multimodal fusion experiment. The overall recognition accuracy for text increased by 6.70% when compared to a single modal, and the speech emotion recognition accuracy increased by 13.85%. The experimental findings demonstrate that multimodal detection accuracy in the test dataset is superior to single-modal detection accuracy and superior to other published multimodal models.

Thaweesak Yingthawornsuk [15] proposed MFCC, which are taken from spoken speech signals, are a method for speech recognition. Prior to training and testing speech samples using maximum likelihood classifiers (ML) and support vector machines (SVM), principal component analysis is used as a supplement to the feature dimensionality reduction state. Based on a database of 40 spoken

utterances from an experimental study that was recorded in acoustically controlled spaces, SVMs containing more randomized MFCC samples were selected from the database, 16 ordered MFCC extract clearly showed an improvement in recognition rate. were compared with ML.

A SER system based on several classifiers and feature extraction techniques has been created, according to Leila Kerkeni et al. [16]. Speech signals are processed to obtain modulation spectrum (MS) characteristics and mel- frequency cepstral coefficients (MFCC),which are then used to train different classifiers. The most pertinent feature subsets were found using feature selection (FS). For the sentiment categorization challenge, a number of machine learning paradigms were employed. Their effectiveness is then contrasted with that of multivariate linear regression (MLR) and support vector machine (SVM) methods, which are often employed in the field of speech signal emotion identification. As test data sets, the Berlin and Spanish databases are employed. This study demonstrates that when speaker normalization (SN) and feature selection are used, all classifiers in the Berlin database achieve 83% accuracy.

S.Ramesh et al. [26] to establish a new database and identify their emotions, the SAVEE and TESS datasets were combined. Our key goal is to characterize their emotions using this solid dataset. We have suggested a fresh machine learning algorithm for this use. The features from the voice signal datasets are first extracted using Mel-frequency cepstral coefficients. Finally, a mix of the naive Bayes machine learning method and the grey wolf optimizer was suggested for classification. According to the findings, our suggested categorization system performs better than current machine learning.

Yeşím Ülgen Sönmez et al. [28] In their study, a brand-new, less computationally complex SER approach has been created. On the databases RAVDESS, EMO-DB, SAVEE, and EMOVO, this technique known as 1BTPDN is used. The raw audio data is first transformed using a one-dimensional discrete wavelet transform to produce the low-pass filter coefficients. Textural analysis techniques, a one-dimensional local binary pattern, and a one-dimensional local ternary pattern are used to extract the features from each filter. The most prominent 1024 features out of 7680

features are chosen using neighborhood component analysis, and the other features are ignored. These 1024 features are chosen as the input for the classifier, a support vector machine based on a third-degree polynomial kernel. In the databases RAVDESS, EMO-DB, SAVEE, and EMOVO, the success rates of the 1BTPDN were 95.16%, 89.16%, 76.67%, and 74.31%, respectively.

CHAPTER-3
EXISTING SYSTEM

In the existing system of speech emotion recognition, advanced technologies are employed to understand and interpret the emotions conveyed through spoken words. The process involves converting speech signals into visual representations known as spectrograms, capturing the frequency content over time. Machine learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), play a crucial role in this system. These models are trained on large datasets containing diverse speech samples expressing various emotions. During training, the models learn to recognize patterns and associations between acoustic features and specific emotional states. The success of the system lies in its ability to generalize this learning to new, unseen speech data, allowing it to accurately predict the emotional content of spoken words.

3.1 ISSUES IN EXISTING METHODOLOGY

- The system is very static in nature and cannot provide any good performance in real time systems.
- The system is very slow to compare the correlations of the complete dataset with just one audio file.
- Variable length audio files are not understandable.
- Noice interruption cannot be canceled during recording of audio

CHAPTER-4

PROPOSED SYSTEM

4.1 PROPOSED SYSTEM

Our goal of this project is to design and implement a system that can recognize emotions from speech using speech processing techniques. The system architecture consists of several stages. In the preprocessing stage, voice activity detection is performed to remove non-speech segments, followed by feature extraction to extract relevant features such as MFCCs, pitch, energy, and formants. Feature selection techniques like PCA or Information Gain are then applied to choose a subset of features that discriminate well between different emotions. The selected features are used to train a machine learning model, which can be SVMs, CNNs, or LSTM networks, to classify emotions. Post-processing techniques can be applied to refine the classification results based on context or by combining outputs from multiple models. The system requires an appropriate dataset for training and testing, such as IEMOCAP or RAVDESS.

Evaluation metrics like Accuracy, Precision, Recall, and F1-score are used to measure the system's performance for each emotion category. The applications of this system include customer service chatbots that can adjust their responses based on customer emotions, educational software that can adapt to students' emotional states, and healthcare systems that can identify patients at risk of mental health issues. However, there are challenges to overcome, such as intra-speaker and inter-speaker variability in vocal expressions, cultural and contextual ambiguity of emotions, and the limited size and diversity of available datasets. Potential extensions of the system include incorporating other modalities like facial expressions or textual content for improved accuracy, developing personalized emotion recognition models for individual users, and exploring explainable AI techniques to understand the model's decision-making process.

4.2 Advantages of Proposed system

This proposed system of emotion recognition using speech processing offers significant advantages in understanding and responding to human emotions, fostering more empathetic and responsive interactions in various domains and applications.

1. Through sophisticated machine learning models and feature extraction techniques, the system can accurately identify and classify a wide range of emotions expressed through speech, providing valuable insights into the speaker's emotional state.
2. The system can process speech signals in real-time, enabling immediate recognition and response to emotional cues during conversations, customer interactions, or other live interactions.
3. The system provides valuable insights into users' emotional responses, enabling organizations to make informed decisions regarding product development, marketing strategies, customer satisfaction, and employee well-being.
4. Emotion recognition using speech processing holds promise for applications in mental health monitoring and therapy, assisting healthcare providers in assessing patients' emotional well-being and providing timely interventions.

CHAPTER-5
SYSTEM REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

In order for every software application to run properly, it needs to satisfy a lot of functions that are to be deployed in it. These functions are nothing but various operations that are performed in each step while developing the application. This step comes under the best practices of developing an application. Functional and Non-Functional Requirements together set a list of rules that govern the smooth running of an application and it also helps the developer and the user to determine the software and hardware requirements that are needed to run the application.

To perform emotion recognition using speech processing in Python, you can leverage several libraries and frameworks that offer functionalities for audio processing, feature extraction, machine learning, and deep learning. Here are some essential libraries commonly used for emotion recognition:

1. Librosa: Librosa is a popular Python library for audio and music analysis. It provides tools for loading audio files, extracting various features from audio signals (such as MFCCs, chroma features, and spectral features), and performing signal processing tasks.

2. PyAudio: PyAudio is a Python wrapper for PortAudio, which allows Python applications to easily capture and play audio streams. It provides functions for audio input/output, making it useful for recording audio from microphones or audio interfaces.

3. TensorFlow / PyTorch: TensorFlow and PyTorch are powerful deep learning frameworks that offer flexible tools for building and training neural networks. You can use these frameworks to implement and train deep learning models for emotion recognition, such as convolutional neural networks (CNNs) or RNN.

4. scikit-learn: scikit-learn is a comprehensive machine learning library in Python. It

provides implementations of various machine learning algorithms, including support vector machines (SVM), Gaussian mixture models (GMM), and decision trees, which can be used for emotion classification tasks.

5. Keras: Keras is a high-level neural networks API that runs on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit (CNTK). It offers a user-friendly interface for building and training deep learning models, making it suitable for rapid prototyping and experimentation.

6. OpenSMILE: OpenSMILE is a feature extraction tool designed specifically for audio and speech processing tasks. It offers a wide range of features for capturing acoustic characteristics from speech signals and can be integrated into Python workflows using wrappers or APIs.

These libraries provide a robust ecosystem for developing emotion recognition systems using speech processing techniques in Python. Depending on your specific requirements and preferences, you can choose the appropriate combination of libraries to suit your needs.

5.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are used to set conditions monitor the performance characteristic of the application. It describes how a specific function in the application works. They also determine the overall quality of the project and hence it is a very important aspect in any software development process. Non-functional requirements, the emotion recognition system can deliver a robust, efficient, and user-friendly solution that meets the needs of stakeholders and users effectively.

Non-functional requirements for emotion recognition using speech processing encompass aspects related to system performance, security, usability, and maintainability. Here are some key non-functional requirements:

1. **Performance**:The system should process speech signals and recognize emotions in real-time to ensure timely responses.Emotion recognition should be performed with minimal delay to maintain a seamless user experience.The system should be capable of handling increasing amounts of data and user requests without significant degradation in performance.

2. **Accuracy and Reliability**:Emotion recognition algorithms should achieve high levels of accuracy in identifying and classifying emotional states from speech signals.The system should be resilient to variations in speech characteristics, environmental noise, and speaker variability to maintain consistent performance.

3. **Security and Privacy**: Personal speech data used for emotion recognition should be handled securely and in compliance with relevant privacy regulations.Communication channels used for transmitting speech data and emotion recognition results should be encrypted to prevent unauthorized access or tampering.

4**. Usability**:The system should have an intuitive and user-friendly interface for users to interact with and provide feedback.The system should be accessible to users with disabilities, complying with accessibility standards and guidelines.

5. **Maintainability**:

The system architecture should be modular and well-structured, allowing for easy maintenance, updates, and scalability.Comprehensive documentation should be provided to facilitate system maintenance, troubleshooting, and future development efforts.Source code and configuration files should be managed using version control systems to track changes and collaborate effectively.

6**. Portability**:

Platform Independence: The system should be compatible with multiple operating systems and hardware configurations to ensure broad accessibility and deployment flexibility.

5.3 HARDWARE REQUIREMENTS

- Processor :intel corei3
- Hard disk:250GB
- RAM :8GB and good internet connection
- Devices Required :Speaker

5.4 SOFTWARE REQUIREMENTS

- OS :Windows 10
- Programming language :Python
- ANACONDA 3-64bit
- Jupyter Notebook

5.5 Libraries used

Librosa- A Python library for analyzing music and audio is called librosa.

Matplotlib- A cross-platform library for data visualization and graphical charting is called matplotlib.

TensorFlow- TensorFlow is a Python-friendly open-source toolkit for mathematical computation that accelerates and simplifies AI.

Keras- An open-source programming framework called keras provides fake brain organizations with a Python point of contact. Keras functions as the TensorFlow library connecting point intended to provide fast experimentation.

CHAPTER-6

TECHNOLOGIES ADOPTED

Speech emotion recognition involves the use of various technologies, including signal processing techniques and machine learning algorithms, to analyze and interpret emotional content in spoken language. Here are some key technologies adopted for speech emotion recognition:

1. **Signal Processing**:

Spectrogram Analysis: Converting speech signals into spectrograms, which visually represent the frequency content over time, is a common signal processing technique. Spectrogram analysis provides a foundation for extracting features related to pitch, intensity, and other acoustic characteristics.

2.**Feature Extraction**:

Mel-Frequency Cepstral Coefficients (MFCCs): Extracting MFCCs is a prevalent technique for capturing the spectral characteristics of speech. MFCCs represent the energy distribution across different frequency bands and are crucial features for emotion recognition.

Prosodic Features: Extracting features related to speech prosody, including pitch, rhythm, and speech rate, provides information about the expressive elements of speech associated with emotions.

3.**Machine Learning Algorithms**:

Convolutional Neural Networks (CNNs): CNNs are effective in capturing spatial patterns in spectrograms, making them suitable for extracting emotional features from visual representations of speech.

Recurrent Neural Networks (RNNs): RNNs are designed to recognize temporal patterns in sequences, allowing them to capture the sequential nature of speech and understand how emotional content evolves over time.

4.**Natural Language Processing (NLP):**

Sentiment Analysis Techniques: In some cases, sentiment analysis techniques from the field of natural language processing are integrated to understand the sentiment conveyed through words, contributing to the overall emotion recognition process.

5.Voice and Speech Analytics Platforms:

Commercial Platforms: Various commercial platforms and APIs offer pre-built models and tools for speech emotion recognition. These platforms often leverage a combination of signal processing and machine learning techniques for accurate and efficient emotion analysis.

6.Edge Computing:

On-Device Processing: With the advancement of edge computing, some speech emotion recognition models are implemented to run directly on devices, allowing real-time analysis without the need for continuous internet connectivity.

CHAPTER 7
SYSTEM DESIGN

7.1 ALGORITHM

7.1.1 Support Vector Machines (SVM)

SVM is a supervised learning algorithm that can be used for classification tasks, including emotion recognition. SVMs work well with high-dimensional data, making them suitable for processing features extracted from speech signals.

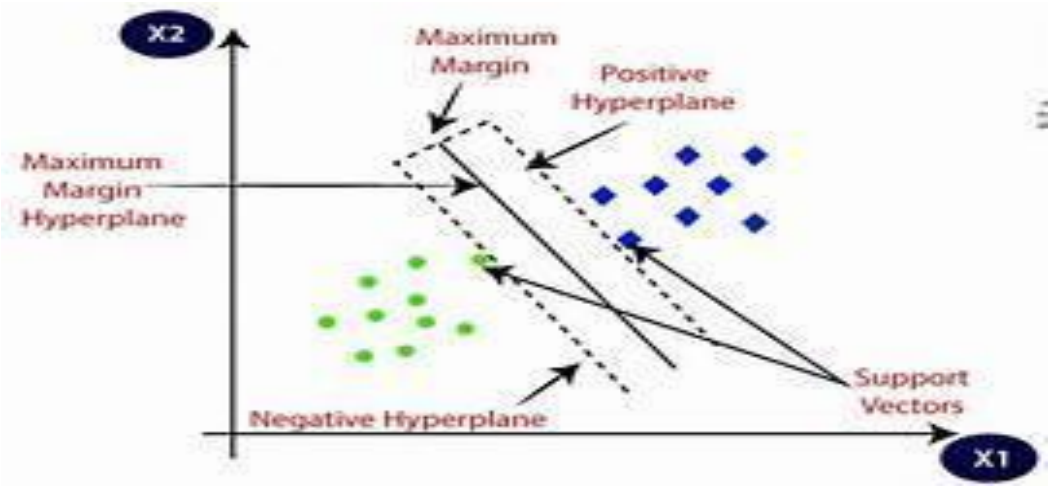


Fig-7.1 Support Vector Machine

7.2 MODULES

1. Data Collection
2. Preprocessing
3. Feature Extractiony
4. Feature Selection
5. Model Training
6. Model Evaluation
7. Model Optimization
8. Integration and deployment

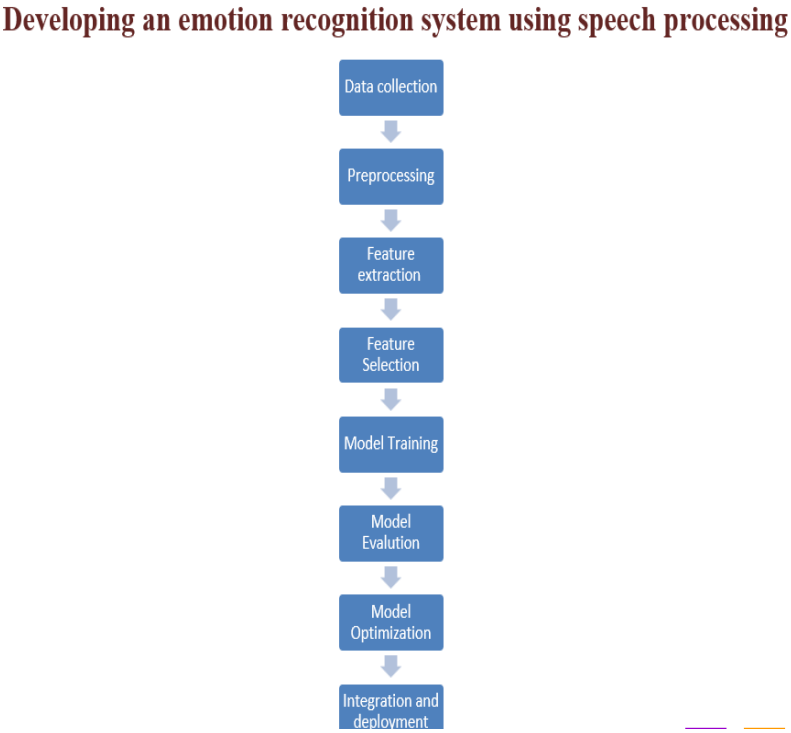


Fig-7.2 Block Diagram of the Application

7.3 UNIFIED MODELING LANGUAGE

UML is a modeling language which is used to visualize a way the software is has developed. It is a visual language which consists of series of steps in which the application works. The UML diagrams depict the structure and behavior of the software application. Visualizing a project before developing an application makes the job easier. It can be used to make a list of requirements for software development. Also, a lot of time is saved where there is a definite model to look up to while developing a software.

UML Diagrams can be broadly classified into:

Behavior diagrams: They capture or visualize the behavior of the diagram. They depict how the behavior changes w.r.t each task or function. These diagrams include Interaction diagrams, Use-case diagrams, State diagrams, Activity Diagrams etc.

Structural diagrams: They concentrate on the structure of the application rather than the functionalities. These diagrams include Class Diagrams, Deployment diagrams, Component diagrams etc.

7.3.1 USE CASE DIAGRAM

They are generally used to visualize the various tasks that are performed in the application. They also depict the different users who have access to perform these tasks. Use-case diagrams come under behavior diagrams because of its emphasis on the tasks performed and the users(actors) who perform these tasks.

The various tasks that are performed in the application is

1.

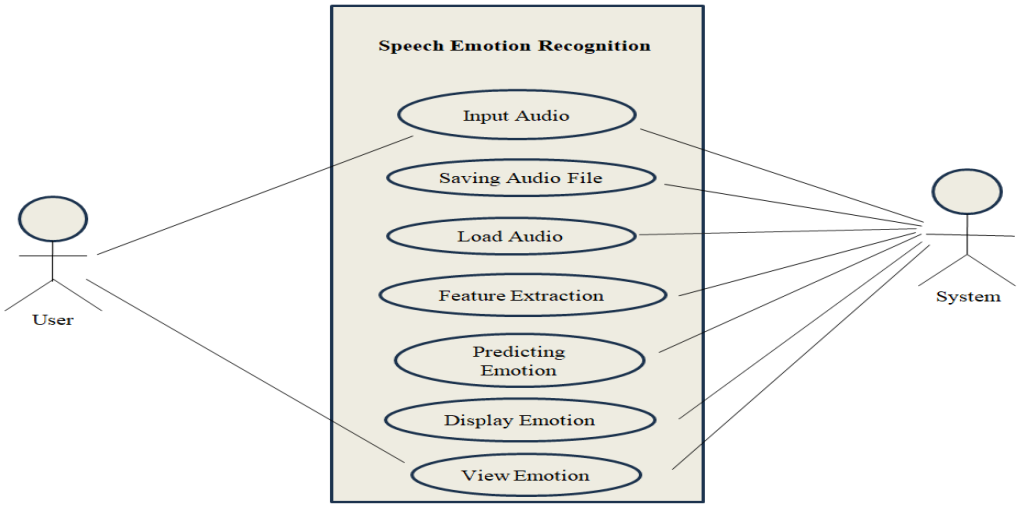


Fig -7.3 USE CASE DIAGRAM FOR SPEECH RECOGNITION

7.3.2 ACTIVITY DIAGRAM

Activity diagram depicts the flow or the sequence of control in an application. We can actually use the activity diagram to verify every task that is performed in the use-case diagram. It also depicts the steps of execution. They basically depict the workflows in the software application. It also emphasizes on the sequence of tasks and the conditions that are to be met in order for

a particular task to perform. This way, it gives us information about what causes a particular task to happen. So, this gives us a high-level visualization of the application.

The main objectives are

1. Depict the activity flow
2. Describe the sequence (branched or sequential)

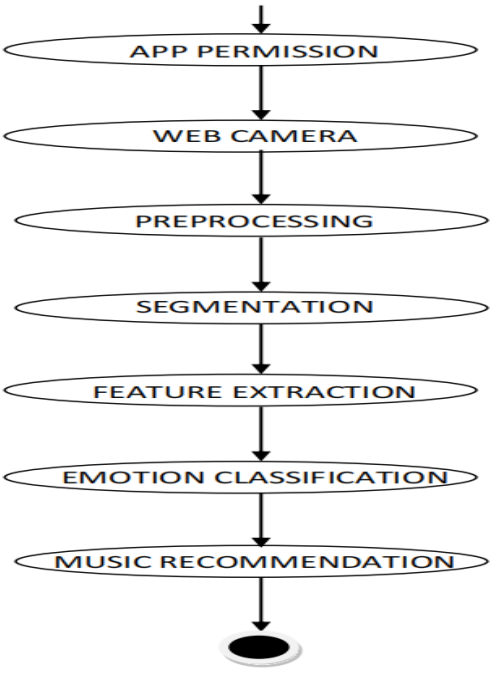


Fig -7.4 ACTIVITY DIAGRAM

7.3.3 SEQUENCE DIAGRAM

It is used to depict the interaction among the objects in an application. They also describe the changes in the behavior of these objects after they interact with each other. They also describe the order in which these interactions take place. Also, developers use these diagrams to make note of all the requirements that are to be needed in order to develop the application.

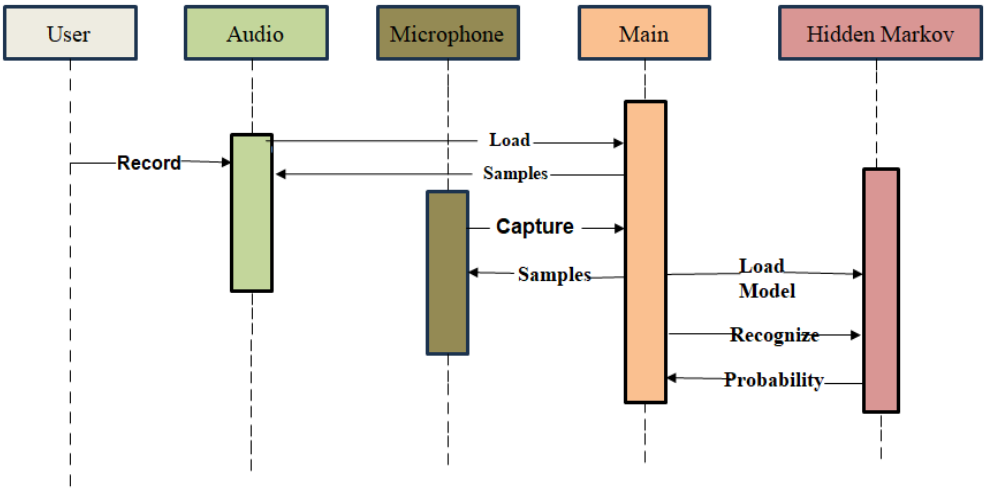


Fig -7.5 SEQUENCE DIAGRAM

CHAPTER-8
SYSTEM IMPLEMENTATION

8.1 SUPPORT VECTOR MACHINE

Speech emotion recognition involves the use of various algorithms and techniques to analyze and classify emotional states based on speech signals. One of the algorithms that plays a vital role in this speech recognition is the support vector machine. SVM is a supervised Machine Learning algorithm .

SVM is a supervised learning algorithm that is widely used for classification tasks, including emotion recognition. It works by finding the hyperplane that best separates different classes in feature space.

8.2 Techniques and algorithms

Speech recognition, also known as automatic speech recognition (ASR), involves converting spoken language into written text. The process typically includes several stages, and various algorithms and techniques are employed. Here's a simplified overview of how speech recognition can be done.

There are several techniques involved in the speech recognition like :

Preprocessing :

Audio Recording: Begin by capturing the spoken audio using a microphone or other recording devices.

Sampling: The continuous audio signal is sampled into discrete points, typically at a rate of 8,000 to 16,000 samples per second.

Noise Reduction: Remove background noise and filter the audio signal to enhance the quality of speech.

Feature Extraction:

MFCC (Mel-Frequency Cepstral Coefficients): Extract features that represent the spectral characteristics of the speech signal. MFCCs are commonly used for this purpose.

Energy and Pitch Features: Include measures of energy and pitch, which convey information about loudness and fundamental frequency.

Acoustic Modeling:

Deep Learning Models: Modern approaches, such as deep neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN), have shown success in acoustic modeling.

Language Modeling:

N-gram Models: Use N-gram language models to estimate the likelihood of word sequences based on training data.

Statistical Language Models: Employ statistical techniques to model the probability of word sequences.

Decoding:

Viterbi Algorithm: Apply the Viterbi algorithm to find the most likely sequence of words given the acoustic model and language model probabilities.

Beam Search: A heuristic search algorithm used to explore multiple hypotheses efficiently.

Post-Processing:

Language Model Integration: Combine the output of the acoustic model with the language model probabilities to improve transcription accuracy.

Evaluation and Refinement:

Evaluation Metrics:Assess the performance of the system using metrics like Word Error Rate (WER) or Character Error Rate (CER).

Integration:

Application Integration: Integrate the speech recognition system into specific applications such as virtual assistants, transcription services, or voice-activated devices.

It's important to note that advancements in deep learning, particularly end-to-end models, have led to significant improvements in speech recognition accuracy. Additionally, the availability of large labeled datasets and powerful computing resources has played a crucial role in the success of modern speech recognition systems.

8.3 DATASETS

Using real-world datasets in speech emotion recognition is crucial for several reasons, as it contributes to the development and improvement of robust and effective models.

Real-world datasets capture a wide range of emotions expressed by individuals in authentic settings. Emotions are complex and context-dependent, and using diverse datasets helps models generalize better to various emotional expressions.

These datasets provide naturalistic expressions of emotions, reflecting how people express themselves in daily life. This helps models better understand and interpret the nuances of genuine emotional communication.

Including diverse speakers and demographic groups in datasets ensures that models can recognize emotions across different individuals, age groups, genders, and cultural backgrounds. This diversity contributes to the model's inclusivity and applicability across various user populations.

Contextual Information:

Real-world datasets often include contextual information surrounding the emotions expressed, providing additional cues for the model. Understanding the context in which emotions occur can improve the model's ability to accurately recognize and interpret emotional states.

Real-world datasets introduce challenges such as noisy and imperfect data. Dealing with these challenges forces models to become more robust and capable of handling imperfect conditions commonly encountered in real-life scenarios.

Ethical Considerations:

Real-world datasets help address ethical considerations related to bias and fairness in emotion recognition. By including diverse samples, models can learn to recognize and respond to emotions without perpetuating biases that may exist in more limited or biased datasets.

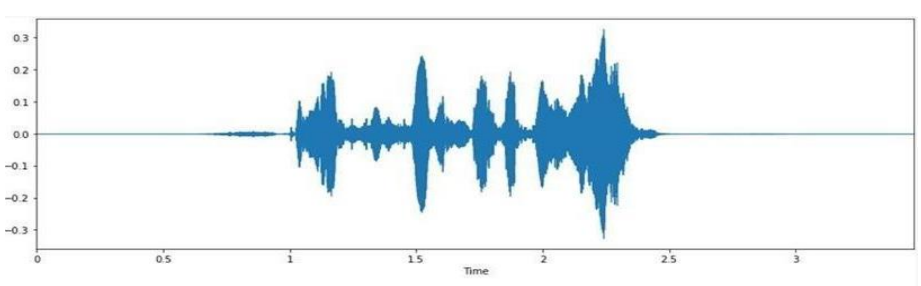
Adaptation to Unforeseen Situations:

Real-world datasets prepare models for unforeseen situations and unexpected emotional expressions. This adaptability is essential for applications like virtual assistants, customer service, or healthcare, where users may express a wide range of emotions.

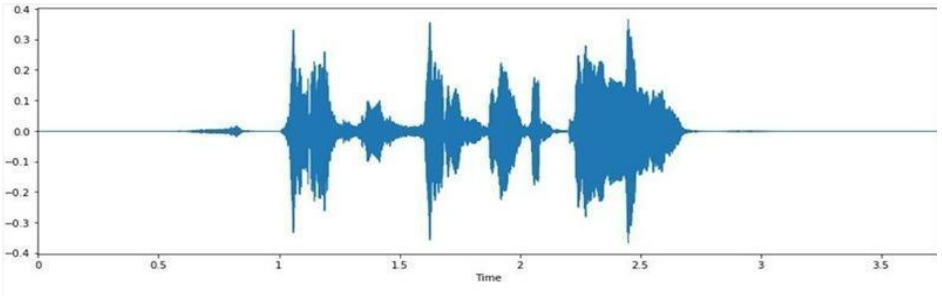
8.3.1 RAVEDESS DATASET

The RAVDESS, Emotional Audio Database at Ryerson University (RAVDESS). There are 7356 files in the (RAVDESS) (total size: 24.8 GB). 24 professional actors:12 male and 12 female—perform 2 lexically similar phrases in the database with neutral American accents. Both in speech and in the lyrics, there are expressions of calmness, happiness, joy, sadness, anger, fear, surprise, and contempt.

There are two emotional intensity levels (normal and strong) and one neutral expression created for each expression. Three modality forms are accessible for all conditions: Audio-only (16bit, 48kHz.wav), Audio-Video (720p H.264, AAC 48kHz,.mp4), and Video-only (480p H.264, AAC 48kHz,.mp4) (no sound).



-Graph-1: Wave Plot Of Fearful Audio Of RAVDNESS Dataset



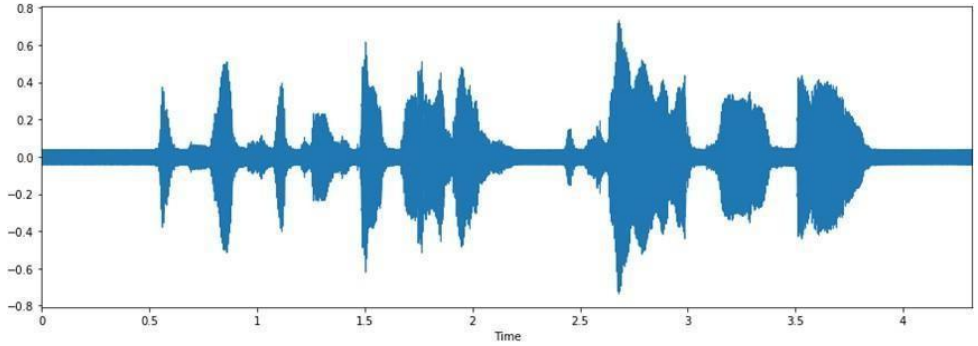
Graph-2: Wave Plot Of Happy Audio Of RAVDNESS Dataset

8.3.2 SAVEE

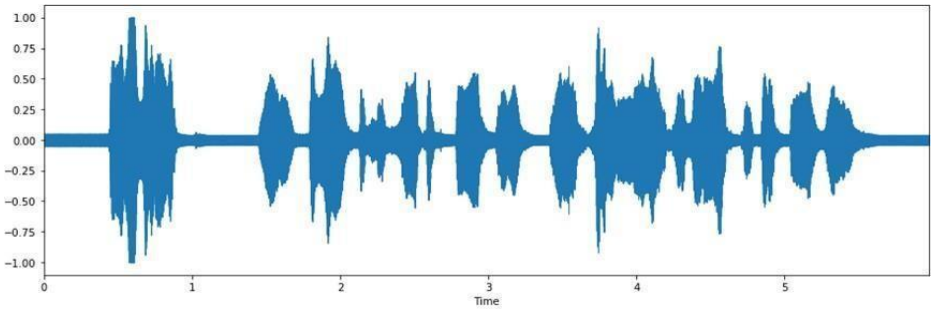
Surrey General media Communicated Feeling (SAVEE) information base has been recorded as a pre-imperative for the improvement of a programmed feeling acknowledgment framework. The data set comprises accounts from 4 male entertainers in 7 distinct feelings, 480 English expressions altogether. The sentences were looked over the standard TIMIT corpus and phonetically adjusted for every inclination. The information was kept in a visual media lab with great general media gear, handled and marked. To check the nature of execution, the accounts were assessed by 10 subjects under sound, visual and general media conditions.

Characterization frameworks were assembled involving standard highlights and classifiers for every one of the sound, visual and general media modalities, and speaker-free acknowledgment paces of 61%, 65% and 84% accomplished separately.

The first source includes four folders, each addressing a speaker. However, I combined all of them into a single organizer, so the first two letters of the filename are the initials of each speaker. For instance, "DC d03.wav" is the speaker DC's third expression of nausea. The fact that they are male speakers is meaningless. This won't be a problem since the TESS dataset, which is almost all female, will balance things out.



Graph 1: Wave plot of fearful audio of SAVEE dataset



Graph 2: Wave plot of happy audio of SAVEE dataset

8.3.3 TESS

One of the four important datasets that I was fortunate to discover is the (TESS) dataset. It's Intriguing that this dataset solely includes females and yet the audio is of such good caliber. The other sample is primarily composed of male speakers, creating a slightly unbalanced representation. Consequently, in terms of generalization, this dataset would serve as a very good training dataset for the emotion classifier (not overfitting).

Two actresses (26 and 64 years old) recited a set of 200 target words in the carrier phrase "Say the word _," and recordings of the set evoking each of the seven emotions were created.There are a total of 2800 audio files.

Each of 2 female performers and their emotions are included within an own folder in the dataset, which is organized this way. Additionally, all 200 target word audio files are contained within that.

8.4 Methodology

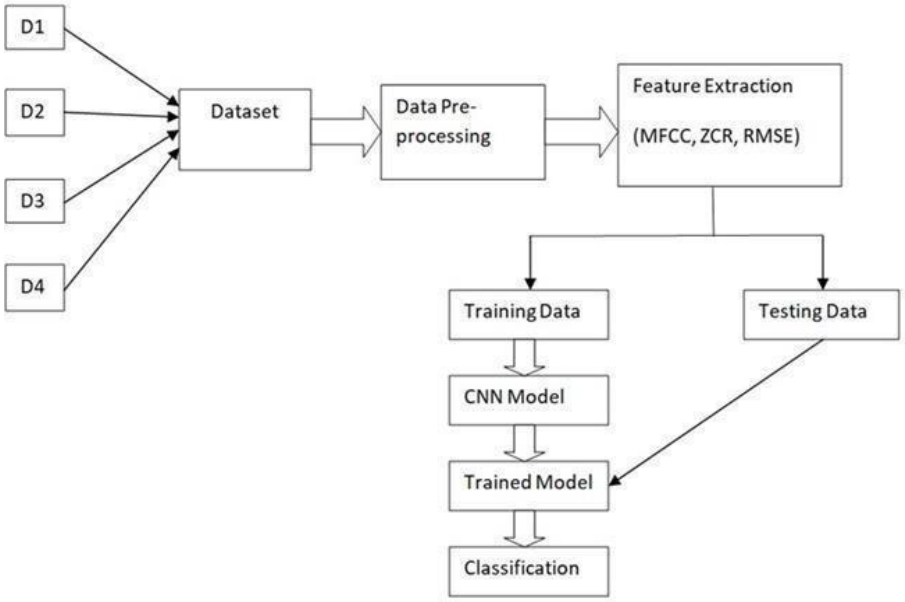


Fig-8.1 Flow Chart of Speech Emotion Recognition

8.4.1 Data pre-processing

Data preprocessing plays a crucial role in speech emotion recognition as it involves transforming raw speech data into a format that is suitable for analysis and model training.

Data preprocessing is a pivotal step in the realm of speech emotion recognition, involving the transformation of raw audio data into a format conducive to analysis and model training. The process begins with the initial capture of audio data using microphones or recording devices, followed by the crucial step of sampling, where the continuous analog signal is discretized into discrete points at a predefined rate, typically ranging from 8,000 to 16,000 samples per second. This digitization lays the foundation for subsequent digital processing.

Mel-Frequency Cepstral Coefficients stand out as a prevalent technique for capturing the spectral characteristics of speech, providing valuable information for recognition. Additional features,like delta delta-delta coefficients, along with energy and pitch features, are computed to encapsulate dynamic aspects and variations in loudness and fundamental frequency.

Normalization is a subsequent step, ensuring that extracted features maintain similar scales. Techniques like z-score normalization or min-max scaling are applied to prevent certain features from dominating others during model training. Segmentation divides speech signals into short frames, typically lasting 20-30 milliseconds, to capture temporal variations. Overlapping frames are employed to prevent information loss at frame boundaries. Windowing, achieved through functions like Hamming or Hanning windows, reduces spectral leakage and smooths the signal, facilitating frequency domain analysis.

Data augmentation is a valuable technique, introducing variations to the dataset to increase diversity. This can involve the introduction of background noise, changes in pitch or speed, or alterations to other acoustic characteristics. Handling imbalanced datasets is crucial, with preprocessing techniques addressing class imbalance to prevent model bias towards the majority class.

Considerations for different languages and accents may also necessitate preprocessing steps, including language identification and accent normalization, ensuring consistency in feature representations. Quality control is applied to identify and eliminate low-quality recordings or instances with poor signal-to-noise ratios, maintaining the overall quality of the dataset.

8.4.2 Feature extraction

Feature extraction assists with lessening how much excess information from the informational index. Extraction of highlights is a vital part in dissecting and tracking down relations between various things. The information given of sound can't be perceived by the models straightforwardly to change over them into a reasonable organization highlight extraction is utilized.**Feature extraction** is a fundamental and critical step in speech emotion recognition, serving as the process by which raw speech signals are transformed into a set of discerning features that encapsulate the distinct characteristics associated with diverse emotional states. The importance of feature extraction lies in its ability to capture acoustic attributes, providing a nuanced representation of the underlying emotional expressions within speech

MFCC

MFCC is typically used as components in discourse recognition frameworks, such as those that can recognise numbers spoken into a phone. MFCCs are also increasingly keeping track of participation in applications for music data recovery, such as kind categorization, sound closeness measurements, and so forth.

Mel-Frequency Cepstral Coefficients (MFCCs), a prevalent technique, succinctly encapsulate the spectral content of speech by mathematically transforming the energy distribution across different frequency bands. This compact representation is instrumental in conveying the spectral features crucial for emotional expression. In addition to static features like MFCCs, dynamic features such as **delta and delta-delta coefficients** are extracted to account for temporal variations in emotional speech.

Furthermore, features related to **energy and pitch** are extracted to enrich the model's understanding of loudness and fundamental frequency, both of which can be indicative of specific emotional states. Formant frequencies, representing resonant frequencies in the speech signal and associated with the vocal tract's shape, offer insights into vowel sounds, adding another layer of detail to the emotional context. **Prosodic features**, encompassing aspects of speech rhythm, intonation, and tempo, are considered, capturing the rhythmic nuances inherent in emotional speech.

In the realm of statistical descriptors, extracting measures like mean, standard deviation, skewness, and kurtosis provides a concise summary of the distribution of acoustic features. These descriptors offer a compact representation of the statistical properties that contribute to the emotional characteristics of the speech signal. With the advent of deep learning,

features can now be directly extracted from the spectrogram using deep neural networks, allowing for the automatic learning of hierarchical representations from raw speech data. This advancement leverages the power of deep learning architectures to enhance the model's capability in recognizing complex emotional patterns.

Additionally, **contextual information**, including details about the speaker, the conversation, or the environment, may be incorporated as features. This contextual information provides supplementary cues for understanding the emotional context of speech, contributing to a more comprehensive and nuanced model.

8.4.3 Training Data and Testing Data

In speech emotion recognition, the division of data into training and testing sets is a critical aspect of the model development process. The objective is to train the model on a subset of the data and then evaluate its performance on a separate, unseen subset to assess its generalization capabilities. The process typically involves a careful partitioning of the dataset to ensure that the model learns from a diverse range of examples and can effectively handle new data.

Training Data:

The training data constitute the portion of the dataset used to teach the model the underlying patterns and features associated with different emotional states in speech. During this phase, the model learns to map the input speech features to corresponding emotional labels through iterative optimization. The training set should be sufficiently large and diverse, encompassing a variety of speakers, emotional expressions, and acoustic conditions to promote robust learning. A common split is to allocate around
70-80% of the dataset to training, ensuring that the model has ample examples to learn from and can generalize well to unseen instances.

Testing Data:
The testing data, on the other hand, serve as a means to evaluate the model's performance on unseen instances. This subset is crucial for assessing how well the model can generalize its learned patterns to new data, simulating real-world scenarios where the model encounters speech signals it has not encountered during training. Typically, 20-30% of the dataset is reserved for testing to provide a substantial and representative evaluation. The testing set should ideally reflect the same diversity of speakers, emotional expressions, and acoustic conditions present in real-world applications.

It is essential to avoid data leakage between the training and testing sets,
ensuring that instances in the testing set are entirely independent of those used for training. This prevents the model from memorizing specific instances and encourages it to learn underlying patterns that can be applied to new, unseen data. The partitioning of data into training and testing sets is a crucial step in the model development process, contributing to the robustness and reliability .

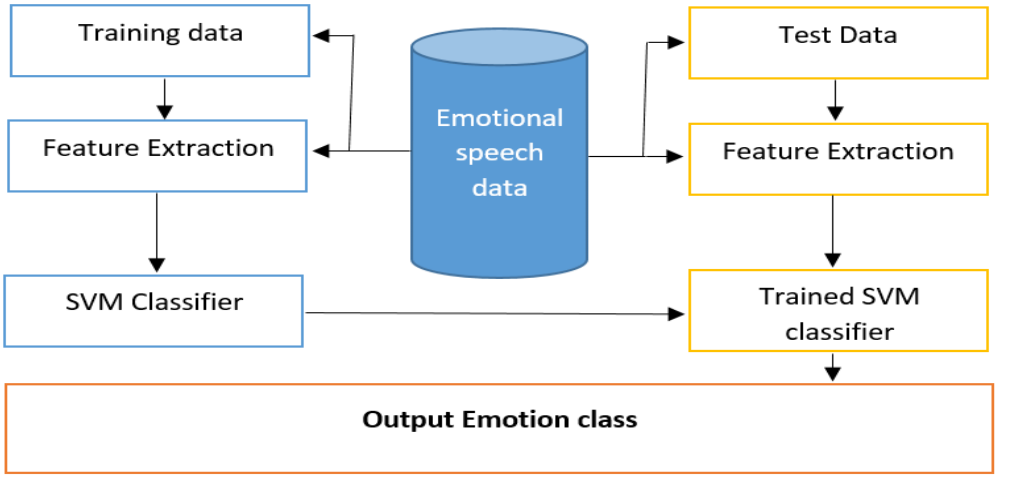


Fig-8.2 Training Data and Testing Data

8.4.4 Convolutional Neural Networks Implementation

One of the most convincing developments in computer vision has been the use of convolutional neural networks. They have performed significantly better than typical PC vision and have produced cutting-edge outcomes. These neural networks have demonstrated real success across a wide range of actual contextual studies and applications.

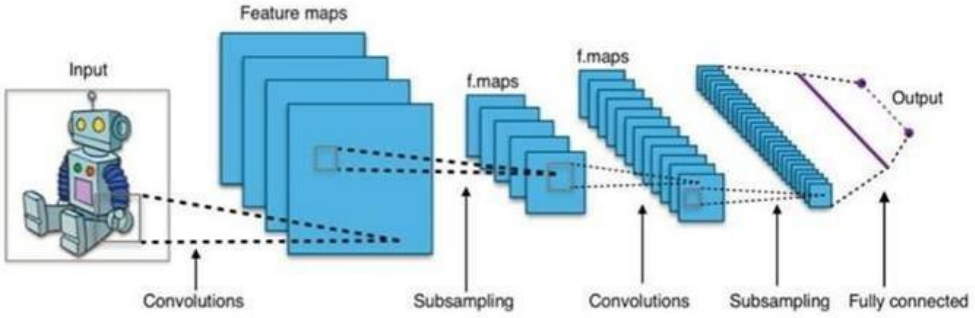


Fig-8.3 : General Architecture of Convolutional Neural Network

Convolutional Neural Networks (CNNs) are like super-smart detectives for understanding emotions in speech. Imagine your speech turned into a colorful picture called a spectrogram, showing how different sounds happen over time. CNNs are great at looking at different parts of this picture and figuring out special patterns that tell them about emotions. They have a cool trick called local receptive fields, which helps them focus on small parts of the picture at a time. It's like they're checking specific areas to find important clues about how you're feeling. These networks learn step by step, starting with basic features like edges and shapes and combining them to understand complex patterns, kind of like building blocks.

CNNs are also good at spotting emotions no matter where they show up in the picture. They don't get confused if the emotional clues appear in different spots or at different times. This makes them great for figuring out the feelings expressed in speech, which often change over time. They're like detectives that don't miss a beat! And guess what? CNNs can work with all kinds of speech pictures, even if they're a bit different or come from different situations. They're like detectives that can adapt to any mystery.

Think of them teaming up with other clever networks, like Recurrent Neural Networks (RNNs), which help them understand the sequence of sounds and emotions. Together, they're like a dynamic duo, capturing the rhythm of speech and making sure nothing emotional goes unnoticed. Sometimes, they even borrow knowledge from their friends who are good at recognizing things in pictures to get even better at understanding emotions in speech. In simple terms, CNNs are like smart detectives trained to look at colorful speech pictures and figure out the emotions behind them. They're great at spotting patterns and adapting to different situations, making them key players in understanding how we express feelings through speech.

8.4.5 Python Libraries

Librosa- A Python library for analyzing music and audio is called librosa. **NumPy**- A Python package used for cluster operations is called numPy. It can also operate in the areas of straight polynomial math, fourier transform, and grids.

Matplotlib- A cross-platform library for data visualization and graphical charting is called matplotlib.

TensorFlow- TensorFlow is a Python-friendly open-source toolkit for mathematical computation that accelerates and simplifies AI.

Keras- An open-source programming framework called keras provi+des fake brain organizations with a Python point of contact. Keras functions as the TensorFlow library connecting point intended to provide fast experimentation.

8.5 CODE IMPLEMENTATION

```
import pandas as pd
import numpy as np

import os
import sys

# librosa is a Python library for analyzing audio and music. It can be used to extract the
    data from the audio files we will see later.
import librosa
import librosa.display
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import train_test_split

# to play the audio files
from IPython.display import Audio
import keras
from keras.callbacks import ReduceLROnPlateau
from keras.models import Sequential
from keras.layers import Dense, Conv1D, MaxPooling1D, Flatten, Dropout,

BatchNormalization
from keras.utils import np_utils, to_categorical
```

```
from keras.callbacks import ModelCheckpoint

import warnings
if not sys.warnoptions:
warnings.simplefilter("ignore")
warnings.filterwarnings("ignore", category=DeprecationWarning)

# Paths for data.
Ravdess = "/kaggle/input/ravdess-emotional-speech-audio/audio_speech_actors_01-24/"
Crema = "/kaggle/input/cremad/AudioWAV/"
Tess = "/kaggle/input/toronto-emotional-speech-set-tess/tess toronto emotional speech set data/TESS Toronto emotional speech set data/"
Savee = "/kaggle/input/surrey-audiovisual-expressed-emotion-savee/ALL/"
ravdess_directory_list = os.listdir(Ravdess)

file_emotion = []
file_path = []
for dir in ravdess_directory_list:
# as there are 20 different actors in our previous directory we need to extract files for each actor.
actor = os.listdir(Ravdess + dir)
for file in actor:
part = file.split('.')[0]
part = part.split('-')
# The third part in each file represents the emotion associated with that file.

file_emotion.append(int(part[2]))
```

```
file_path.append(Ravdess + dir + '/' + file)

# dataframe for emotion of files
emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])

# dataframe for path of files.
path_df = pd.DataFrame(file_path, columns=['Path'])
Ravdess_df = pd.concat([emotion_df, path_df], axis=1)

# changing integers to actual emotions.
Ravdess_df.Emotions.replace({1:'neutral', 2:'calm', 3:'happy', 4:'sad', 5:'angry',
6:'fear', 7:'disgust', 8:'surprise'}, inplace=True)
Ravdess_df.head()
ravdess_directory_list = os.listdir(Ravdess)

file_emotion = []
file_path = []
for dir in ravdess_directory_list:
# as there are 20 different actors in our previous directory we need to extract files for
each actor.
actor = os.listdir(Ravdess + dir)
for file in actor:
part = file.split('.')[0]
part = part.split('-')
# The third part in each file represents the emotion associated with     that file.
file_emotion.append(int(part[2]))

file_path.append(Ravdess + dir + '/' + file)
```

```
# dataframe for emotion of files
emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])

# dataframe for path of files.
path_df = pd.DataFrame(file_path, columns=['Path'])
Ravdess_df = pd.concat([emotion_df, path_df], axis=1)

# changing integers to actual emotions.
Ravdess_df.Emotions.replace({1:'neutral', 2:'calm', 3:'happy', 4:'sad', 5:'angry',
6:'fear', 7:'disgust', 8:'surprise'}, inplace=True)
Ravdess_df.head()
crema_directory_list = os.listdir(Crema)

file_emotion = []
file_path = []

for file in crema_directory_list:
# storing file paths
file_path.append(Crema + file)
# storing file emotions
part=file.split('_')
if part[2] == 'SAD':
file_emotion.append('sad')
elif part[2] == 'ANG':
file_emotion.append('angry')
elif part[2] == 'DIS':
file_emotion.append('disgust')
```

```
elif part[2] == 'FEA':
file_emotion.append('fear')
elif part[2] == 'HAP':
file_emotion.append('happy')
elif part[2] == 'NEU':
file_emotion.append('neutral')
else:
file_emotion.append('Unknown')

# dataframe for emotion of files
emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])

# dataframe for path of files.
path_df = pd.DataFrame(file_path, columns=['Path'])
Crema_df = pd.concat([emotion_df, path_df], axis=1)
Crema_df.head()
tess_directory_list = os.listdir(Tess)

file_emotion = []
file_path = []

for dir in tess_directory_list:
directories = os.listdir(Tess + dir)
for file in directories:
part = file.split('.')[0]
part = part.split('_')[2]
if part=='ps':
file_emotion.append('surprise')
```

```
else:
file_emotion.append(part)
file_path.append(Tess + dir + '/' + file)

# dataframe for emotion of files
emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])

# dataframe for path of files.
path_df = pd.DataFrame(file_path, columns=['Path'])
Tess_df = pd.concat([emotion_df, path_df], axis=1)
Tess_df.head()
savee_directory_list = os.listdir(Savee)

file_emotion = []
file_path = []

for file in savee_directory_list:
file_path.append(Savee + file)
part = file.split('_')[1]
ele = part[:-6]
if ele=='a':
file_emotion.append('angry')
elif ele=='d':
file_emotion.append('disgust')
elif ele=='f':
file_emotion.append('fear')
elif ele=='h':
file_emotion.append('happy')
```

```
elif ele=='n':
file_emotion.append('neutral')
elif ele=='sa':
file_emotion.append('sad')
else:
file_emotion.append('surprise')

# dataframe for emotion of files
emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])

# dataframe for path of files.
path_df = pd.DataFrame(file_path, columns=['Path'])
Savee_df = pd.concat([emotion_df, path_df], axis=1)
Savee_df.head()
# creating Dataframe using all the 4 dataframes we created so far.
data_path = pd.concat([Ravdess_df, Crema_df, Tess_df, Savee_df], axis = 0)
data_path.to_csv("data_path.csv",index=False)
data_path.head()
plt.title('Count of Emotions', size=16)
sns.countplot(data_path.Emotions)
plt.ylabel('Count', size=12)
plt.xlabel('Emotions', size=12)
sns.despine(top=True, right=True, left=False, bottom=False)
plt.show()
def create_waveplot(data, sr, e):
plt.figure(figsize=(10, 3))
plt.title('Waveplot for audio with {} emotion'.format(e), size=15)
librosa.display.waveplot(data, sr=sr)
```

```
plt.show()


def create_spectrogram(data, sr, e):
# stft function converts the data into short term fourier transform
X = librosa.stft(data)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(12, 3))
plt.title('Spectrogram for audio with {} emotion'.format(e), size=15)
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='hz')
#librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='log')
plt.colorbar()
emotion='fear'
path = np.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)


emotion='angry'
path = np.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)
    emotion='sad'
path = np.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
```

```
create_spectrogram(data, sampling_rate, emotion)
Audio(path)
    emotion='happy'
path = np.array(data_path.Path[data_path.Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)

def noise(data):
noise_amp = 0.035*np.random.uniform()*np.amax(data)
data = data + noise_amp*np.random.normal(size=data.shape[0])
return data

def stretch(data, rate=0.8):
return librosa.effects.time_stretch(data, rate)

def shift(data):
shift_range = int(np.random.uniform(low=-5, high = 5)*1000)
return np.roll(data, shift_range)


def pitch(data, sampling_rate, pitch_factor=0.7):
return librosa.effects.pitch_shift(data, sampling_rate, pitch_factor)

# taking any example and checking for techniques.
path = np.array(data_path.Path)[1]
data, sample_rate = librosa.load(path)
```

```
plt.figure(figsize=(14,4))
librosa.display.waveplot(y=data, sr=sample_rate)
Audio(path)
x = noise(data)
plt.figure(figsize=(14,4))
librosa.display.waveplot(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
x = stretch(data)
plt.figure(figsize=(14,4))
librosa.display.waveplot(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
x = shift(data)
plt.figure(figsize=(14,4))
librosa.display.waveplot(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
x = pitch(data, sample_rate)
plt.figure(figsize=(14,4))
librosa.display.waveplot(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
def extract_features(data):
# ZCR
result = np.array([])
zcr = np.mean(librosa.feature.zero_crossing_rate(y=data).T, axis=0)
result=np.hstack((result, zcr)) # stacking horizontally

# Chroma_stft
stft = np.abs(librosa.stft(data))
chroma_stft = np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T, axis=0)
```

```
  result = np.hstack((result, chroma_stft)) # stacking horizontally

  # MFCC
  mfcc = np.mean(librosa.feature.mfcc(y=data, sr=sample_rate).T, axis=0)
  result = np.hstack((result, mfcc)) # stacking horizontally

  # Root Mean Square Value
  rms = np.mean(librosa.feature.rms(y=data).T, axis=0)
  result = np.hstack((result, rms)) # stacking horizontally

  # Mel Spectrogram
  mel = np.mean(librosa.feature.melspectrogram(y=data, sr=sample_rate).T, axis=0)
  result = np.hstack((result, mel)) # stacking horizontally

  return result

  def get_features(path):
  # duration and offset are used to take care of the no audio in start and the ending of each audio file as seen above.
  data, sample_rate = librosa.load(path, duration=2.5, offset=0.6)

  # without augmentation
  res1 = extract_features(data)
  result = np.array(res1)

  # data with noise
  noise_data = noise(data)
  res2 = extract_features(noise_data)
```

```
result = np.vstack((result, res2)) # stacking vertically

# data with stretching and pitching
new_data = stretch(data)
data_stretch_pitch = pitch(new_data, sample_rate)
res3 = extract_features(data_stretch_pitch)
result = np.vstack((result, res3)) # stacking vertically

return result
X, Y = [], []
for path, emotion in zip(data_path.Path, data_path.Emotions):
feature = get_features(path)
for ele in feature:
X.append(ele)
# appending emotion 3 times as we have made 3 augmentation techniques on each audio file.
Y.append(emotion)
len(X), len(Y), data_path.Path.shape
Features = pd.DataFrame(X)

Features['labels'] = Y
Features.to_csv('features.csv', index=False)
Features.head()
X = Features.iloc[: ,:-1].values
Y = Features['labels'].values
# As this is a multiclass classification problem onehotencoder our Y.
encoder = OneHotEncoder()
Y = encoder.fit_transform(np.array(Y).reshape(-1,1)).toarray()
```

```
# splitting data
x_train, x_test, y_train, y_test = train_test_split(X, Y, random_state=0, shuffle=True)
x_train.shape, y_train.shape, x_test.shape, y_test.shape
# scaling our data with sklearn's Standard scaler
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
x_train.shape, y_train.shape, x_test.shape, y_test.shape
# making our data compatible to model.
x_train = np.expand_dims(x_train, axis=2)
x_test = np.expand_dims(x_test, axis=2)
x_train.shape, y_train.shape, x_test.shape, y_test.shape
model=Sequential()
model.add(Conv1D(256, kernel_size=5, strides=1, padding='same', activation='relu',
input_shape=(x_train.shape[1], 1)))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))

model.add(Conv1D(256, kernel_size=5, strides=1, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))

model.add(Conv1D(128, kernel_size=5, strides=1, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))
model.add(Dropout(0.2))

model.add(Conv1D(64, kernel_size=5, strides=1, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))

model.add(Flatten())
```

```
model.add(Dense(units=32, activation='relu'))
model.add(Dropout(0.3))


model.add(Dense(units=8, activation='softmax'))
model.compile(optimizer = 'adam' , loss = 'categorical_crossentropy' , metrics = ['accuracy'])
    model.summary()
rlrp = ReduceLROnPlateau(monitor='loss', factor=0.4, verbose=0, patience=2, min_lr=0.0000001)
history=model.fit(x_train, y_train, batch_size=64, epochs=50, validation_data=(x_test, y_test), callbacks=[rlrp])
print("Accuracy of our model on test data : " , model.evaluate(x_test,y_test)[1]*100 , "%")
   epochs = [i for i in range(50)]
fig , ax = plt.subplots(1,2)
train_acc = history.history['accuracy']
train_loss = history.history['loss']
test_acc = history.history['val_accuracy']
test_loss = history.history['val_loss']
   fig.set_size_inches(20,6)
ax[0].plot(epochs , train_loss , label = 'Training Loss')
ax[0].plot(epochs , test_loss , label = 'Testing Loss')
ax[0].set_title('Training & Testing Loss')
ax[0].legend()
ax[0].set_xlabel("Epochs")
    ax[1].plot(epochs , train_acc , label = 'Training Accuracy')
ax[1].plot(epochs , test_acc , label = 'Testing Accuracy')
ax[1].set_title('Training & Testing Accuracy')
ax[1].legend()
```

```
ax[1].set_xlabel("Epochs")
plt.show()
# predicting on test data.
pred_test = model.predict(x_test)
y_pred = encoder.inverse_transform(pred_test)
      y_test = encoder.inverse_transform(y_test)
df = pd.DataFrame(columns=['Predicted Labels', 'Actual Labels'])
df['Predicted Labels'] = y_pred.flatten()
df['Actual Labels'] = y_test.flatten()
       df.head(10)
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize = (12, 10))
cm = pd.DataFrame(cm , index = [i for i in encoder.categories_] , columns = [i for i in
encoder.categories_])
sns.heatmap(cm, line color='white', cmap='Blues', linewidth=1, annot=True, fmt='')
plt.title('Confusion Matrix', size=20)
plt.xlabel('Predicted Labels', size=14)
plt.ylabel('Actual Labels', size=14)
plt.show()print(classification_report(y_test, y_pred))
```

CHAPTER-9
OBJECTIVES

Emotion Understanding:

The primary objective is to accurately identify and classify the emotional states conveyed through speech. By analyzing acoustic features, such as pitch, intensity, and rhythm, the system aims to discern emotions like happiness, sadness, anger.

Human-Computer Interaction Improvement:

Emotion recognition systems strive to enhance human-computer interaction by enabling machines to respond appropriately to emotional cues in speech. This objective involves developing systems that can adapt their responses based on the detected emotions, leading to more empathetic and engaging interactions.

Personalization and Adaptation:

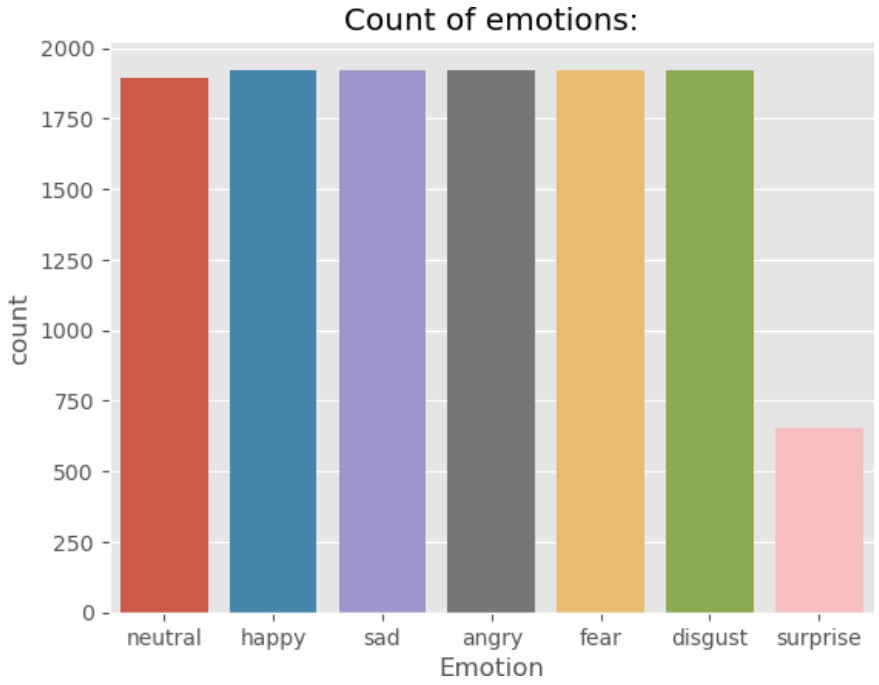
Another goal is to personalize and adapt systems and services based on the emotional state of the user. By understanding the user's emotions, applications can tailor content, recommendations, and responses to better meet individual needs and preferences.

Healthcare Monitoring and Support:

Emotion recognition technology has applications in healthcare for monitoring and supporting emotional well-being. By analyzing changes in speech patterns and emotional expression, these systems can assist in mental health assessments, provide support for individuals with emotional disorders, and contribute to early intervention efforts.

CHAPTER-10
RESULT ANALYSIS

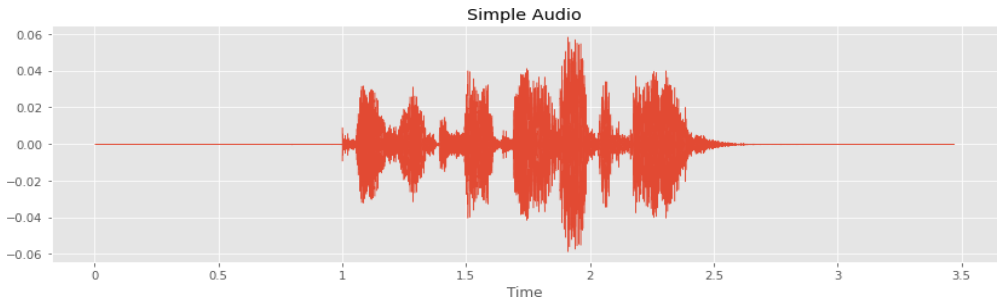
Accordingly, in this model, we obtained model accuracy of around 96% at training data and 71% on the testing data and value loss of about 15% after analysing 48000 samples of data using the MFCC method of feature extraction and the CNN model for training and testing purposes. By using additional feature extraction methods, such as ZCR and RMSE, as well as by providing the model with more data—in our case, we used 48000 samples of data, so the accuracy of the model would increase if the number of samples used were increased. We can further increase the model's accuracy by increasing the number of epochs.



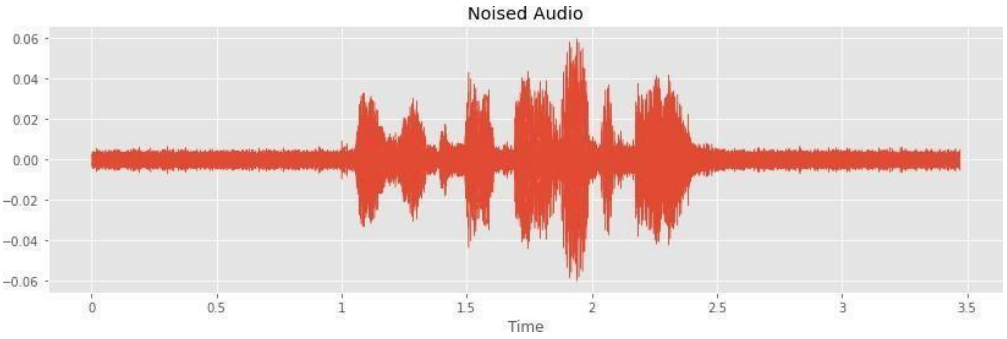
GRAPH-5

The above figure depicts the count values of different emotions in the whole dataset after combining the all four datasets. It is clearly shown that the surprise emotion data points are the least in numbers compared to the other emotion data points.

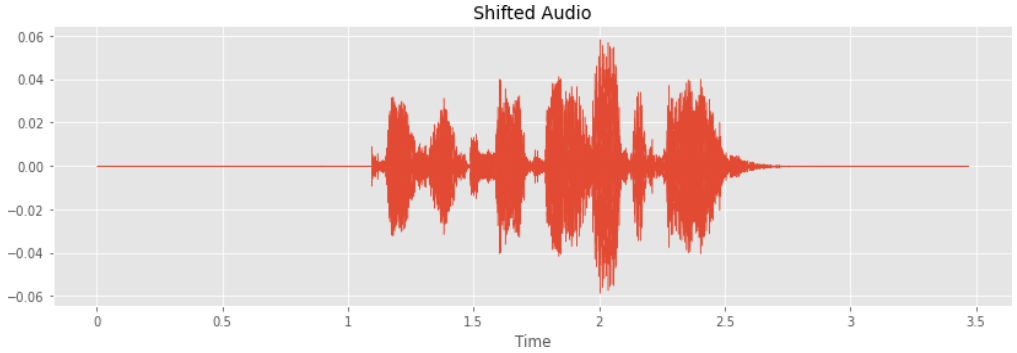
We have also done changes with the audio files by adding noise, elevating the pitch, stretching the audio etc. The following graph shows the different wave plots of the changedaudio samples.



Graph-6



Graph-7



Garph-8

```
In [49]:
model = models.Sequential()
model.add(layers.Conv1D(512, kernel_size=5, strides=1,
                        padding="same", activation="relu",
                        input_shape=(X_train.shape[1], 1)))
model.add(layers.BatchNormalization())
model.add(layers.MaxPool1D(pool_size=5, strides=2, padding="same"))

model.add(layers.Conv1D(512, kernel_size=5, strides=1,
                        padding="same", activation="relu"))
model.add(layers.BatchNormalization())
model.add(layers.MaxPool1D(pool_size=5, strides=2, padding="same"))

model.add(layers.Conv1D(256, kernel_size=5, strides=1,
                        padding="same", activation="relu"))
model.add(layers.BatchNormalization())
model.add(layers.MaxPool1D(pool_size=5, strides=2, padding="same"))

model.add(layers.Conv1D(256, kernel_size=3, strides=1, padding='same', activation='relu'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))

model.add(layers.Conv1D(128, kernel_size=3, strides=1, padding='same', activation='relu'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling1D(pool_size=3, strides = 2, padding = 'same'))

model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dense(7, activation="softmax"))

model.compile(optimizer="rmsprop", loss="categorical_crossentropy", metrics=["acc", f1_m])
```

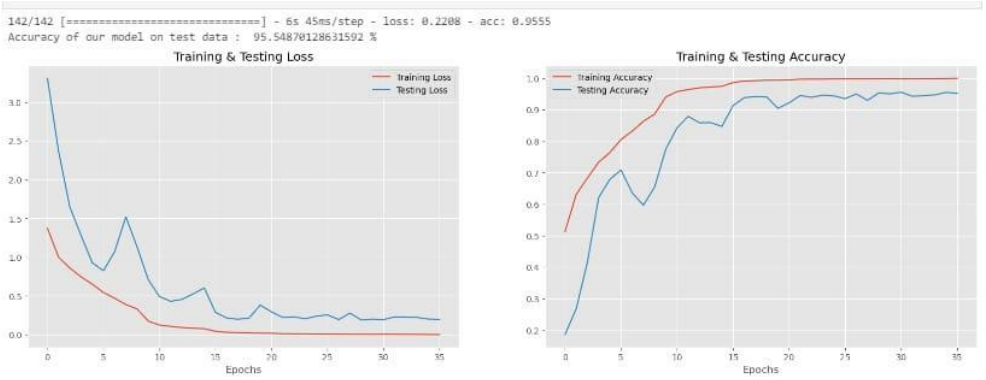
This is basically the CNN function we used for predicting the emotions. There are around 6levels in the model consisting of 3 layers. The layers include the relu function, batch normalisation and max pooling.

```
In [59]:
EPOCHS = 15
batch_size = 32
```

```
In [60]:
history = model.fit(X_train, y_train, validation_data=(X_val, y_val),
                    epochs=EPOCHS, batch_size=batch_size,
                    callbacks=[earlystopping, learning_rate_reduction])
```

```
Epoch 12/15
1095/1095 [==============================] - 122s 111ms/step - loss: 0.1655 - acc: 0.9438 - f1_m: 0.9436 - val_loss: 1.6174
- val_acc: 0.6948 - val_f1_m: 0.6957 - lr: 0.0010
Epoch 13/15
1095/1095 [==============================] - 123s 112ms/step - loss: 0.1521 - acc: 0.9481 - f1_m: 0.9481 - val_loss: 1.5497
- val_acc: 0.7022 - val_f1_m: 0.7025 - lr: 0.0010
Epoch 14/15
1095/1095 [==============================] - 121s 111ms/step - loss: 0.1434 - acc: 0.9519 - f1_m: 0.9521 - val_loss: 1.6115
- val_acc: 0.6984 - val_f1_m: 0.6999 - lr: 0.0010
Epoch 15/15
1095/1095 [==============================] - 120s 109ms/step - loss: 0.1389 - acc: 0.9528 - f1_m: 0.9529 - val_loss: 1.5004
- val_acc: 0.7122 - val_f1_m: 0.7143 - lr: 0.0010
```

We also run around 15 epochs on our model which also helped in increasing the accuracy of the model .



Graph-9 Training testing loss and accuracy

The accuracy of our model is around 96% on the training dataset and 95% on the testing dataset.

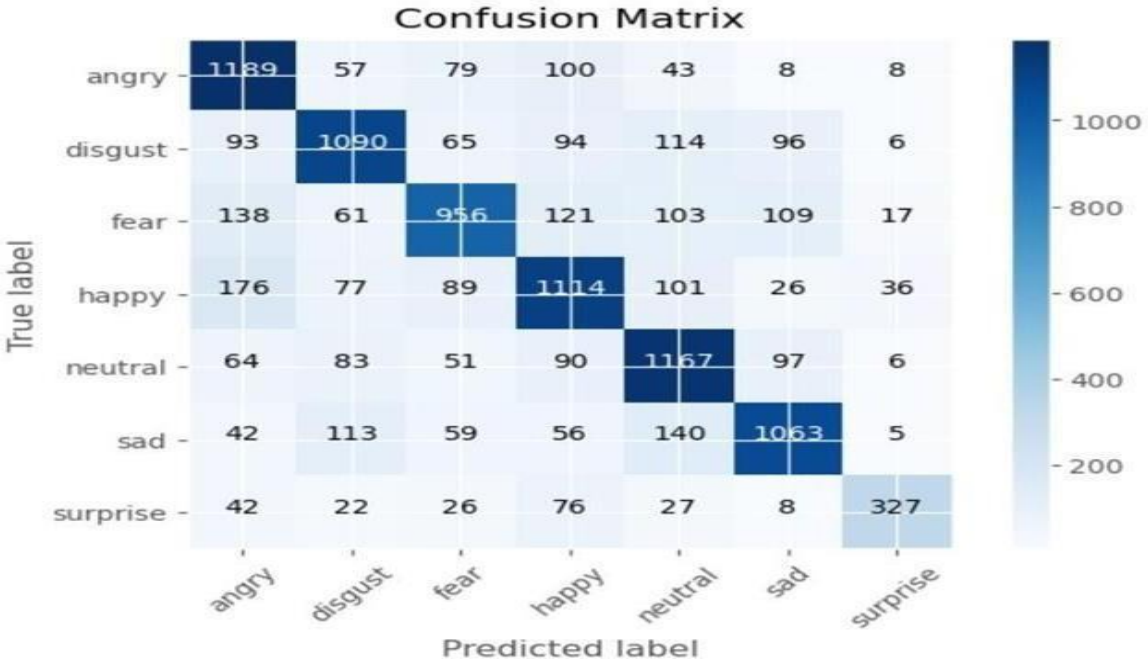


Fig-10.1 Confusion Matrix

CHAPTER-11
CONCLUSIONS

11.1 Conclusions

In the project, we tried to use deep learning to analyze certain speech samples. In order to illustrate the various human emotions, we first loaded the datasets using the Librosa libraryand depicted them in the form of various wave plots and spectrograms. Then, we used the **MFCC feature extraction method** to analyze the acoustic characteristics of all of our samples and organized the sequential data obtained in the 3D array form that the CNN model accepts. Using the **Matplotlib library**, we put the data into a graphical form, then after some repeated Testing with various values reveals that the model's average accuracy is 95% at testing and 75% at the training phase.

11.2 Future Scope

So the discourse feeling acknowledgment is an extremely fascinating subject and there is something else to find in the field, in our model the future work will incorporate the improvement of exactness of the model to come by improved results, we can likewise prepare the model to give aftereffects of the discourse that is longer in term, like in this model we can perceive the feeling just for brief length of time. In future we will be ready to stack the more drawn out example dataset and the model will arrange various feelings in various time frames. Its future work can likewise incorporate the recording of on time information through a receiver with the goal that there is no need of stacking the dataset; we will just train the model and afterward information can be recorded to give the feelings of that individual's voice.

CHAPTER-12

REFERENCES

[1] Ahmed, M. R., Islam, S., Islam, A. K. M. M., & Shatabda, S. (n.d.). *An Ensemble 1D- CNN-LSTM-GRU Model with Data Augmentation for Speech Emotion Recognition*.

[2] Shelke, N., Wadyalkar, V., Kotangale, D., Kuyate, N., Nerkar, A., & Gour, N. (n.d.). A NOVEL APPROACH TO EMOTION DETECTION FROM SPEECH.

[3] Hamsa, S., Shahin, I., Iraqi, Y., & Werghi, N. (2020). Emotion Recognition from Speech Using Wavelet Packet Transform Cochlear Filter Bank and Random Forest Classifier. *IEEEAccess*, *8*, 96994–970006.

[4] Aggarwal, A., Srivastava, A., Agarwal, A., Chahal, N., Singh, D., Alnuaim, A. A., Alhadlaq, A., & Lee, H. N. (2022). Two-Way Feature Extraction for Speech Emotion Recognition Using Deep Learning. *Sensors*, *22*(6).

[5] Hajarolasvadi, N., & Demirel, H. (2019). 3D CNN-based speech emotion recognition using k-means clustering and spectrograms. *Entropy*, *21*(5).

[6] Mustaqeem, Sajjad, M., & Kwon, S. (2020). Clustering-Based Speech Emotion Recognition by Incorporating Learned Features and Deep BiLSTM. *IEEE Access*, *8*, 79861–79875.

[7] Iqbal, A., & Barua, K. (2019). A Real-time Emotion Recognition from Speech using Gradient Boosting. *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 1–5.

[8] Koduru, A., Valiveti, H. B., & Budati, A. K. (2020). Feature extraction algorithms to improve the speech emotion recognition rate. *International Journal of Speech* .

[9] *Speech Emotion Recognition Using CNN Speech Emotion Recognition Using Convolutional Neural Network (CNN) View project Fire Safety in Indian Coal Mines using*

Machine Learning Techniques View project Harini Murugan SRMIST. (n.d.).

[10] Kumar, A., & Iqbal, J. L. M. (2019). Machine Learning Based Emotion Recognition using Speech Signal. *International Journal of Engineering and Advanced Technology (IJEAT)*, *9*, 2249–8958.

[11] Cai, L., Hu, Y., Dong, J., & Zhou, S. (2019). Audio-Textual Emotion Recognition Based On Improved Neural Networks. *Mathematical Problems in Engineering*, *2019*.

[12] Kerkeni, L., Serrestou, Y., Mbarki, M., Raoof, K., Ali Mahjoub, M., & Cleder, C. (2020). Automatic Speech Emotion Recognition Using Machine Learning. In *Social Media And Machine Learning*. IntechOpen.