SPEECH EMOTION RECOGNITION

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

DESAI GANNAMARAJU CHARITH (REG NO: 39110261)
ABHIRAM KANTIPUDI (REG NO: 39110006)

DEPARTMENT OF COMPUTER SCIENCE SCHOOL OF ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with —All grade by NAAC | 12B status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

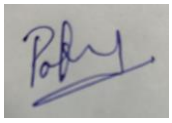
APRIL-2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **DESAI GANNAMARAJU CHARITH (39110261) and ABHIRAM KANTIPUDI (39110006)** who carried out the Project Phase-2 entitled "**SPEECH EMOTION RECOGNITION**" under my supervision from Jan 2023 to April 2023.



Internal Guide

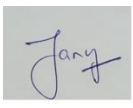
S. POTHUMANI, M.E.

Head of the Department

Dr. L. Lakshmanan, M.E., Ph.D.

Submitted for Viva-voce Examination held on 20.4.2023

Internal Examiner



External Examiner

DECLARATION

We, **DESAI GANNAMARAJU CHARITH (39110261) and ABHIRAM KANTIPUDI (39110006)** hereby declare that the project report entitled "**SPEECH EMOTION RECOGNITION"** done by us under the guidance of **S. POTHUMANI, M.E** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE: 20/04/2023

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATES**

ACKNOWLEDGEMENT

I are pleased to acknowledge my sincere thanks to **the Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

I convey our thanks to **Dr. T. Sasikala, M.E., Ph.D.**, **Dean**, School of Computing, **Dr. L. Lakshmanan, M.E., Ph.D., Head of the Department** of **Computer Science and Engineering,** for providing us with the necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to my project guide, **S. POTHUMANI, M.E.,** for her valuable guidance, suggestions, and constant encouragement, which paved the way for the successful completion of our project work.

I wish to express our thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Emotions play an extremely important role in human mental life. It is a medium of expression of one's perspective or one's mental state to others. Speech Emotion Recognition (SER) can be defined as the extraction of the emotional state of the speaker from his or her speech signal. There are few universal emotions- including Neutral, Anger, Happiness, and Sadness which any intelligent system with finite computational resources can be trained to identify or synthesize as required. In this work, spectral and prosodic features are used for speech emotion recognition because both of these features contain emotional information. Mel-frequency cepstral coefficients (MFCC) are one of the spectral features. Fundamental frequency, loudness, pitch and speech intensity, and glottal parameters are the prosodic features that are used to model different emotions. The potential features are extracted from each utterance for the computational mapping between emotions and speech patterns. Pitch can be detected from the selected features, using which gender can be classified. A support Vector Machine (SVM), is used to classify the gender in this work. Radial Basis Function and Back Propagation Network are used to recognize the emotions based on the selected features and proved that the radial basis function produces more accurate results for emotion recognition than the backpropagation network.

TABLE OF CONTENTS

Chapter No	TITLE		Page No.
	ABSTRACT		v
	LIST OF FIGURES		ix
1	**INTRODUCTION**		1
	1.1	General Information	1
	1.2	Problem statement	3
	1.3	Objectives	4
	1.4	System Architecture	4
	1.5	Statement Scope	6
	1.6	Natural Language Processing	6
2	**LITERATURE SURVEY**		9
	2.2	Open problems in an existing system	10
	2.3	Inferences from the literature survey	12
3	**REQUIREMENT ANALYSIS**		14
	3.1	Feasibility Studies/Risk Analysis of the Project	14
	3.2	Software and Hardware Requirements Specification Document	15
	3.3	System Use case	16
4	**DESCRIPTION OF THE PROPOSED SYSTEM**		17
	4.1	Study of the Project	17
	4.2	Existing Methodology	18

	4.3	Proposed Methodology	23
	4.4	Project Task Set/Project Management Plan	25
5	**IMPLEMENTATION DETAILS**		26
	5.1	Development and Deployment Setup	26
	5.2	Algorithms	29
	5.3	Module Implementation	34
	5.4	Algorithm Used	35
6	**RESULTS AND DISCUSSIONS**		41
7	**CONCLUSION**		45
	7.1	Conclusion	45
	7.2	Future work	46
	7.3	Research issues	47
	7.4	Implementation issues	48
	REFERENCES		49
	APPENDIX		51
	A. SOURCE CODE		51
	B. SCREENSHOTS		62
	C. RESEARCH PAPER		65

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
1.1	System Architecture	4
4.2	Proposed System Architecture	30
4.3	Convolution Neural Network	31
4.4	Pooling Layers	33
4.5	Fully Convolution Layer	35
6.1	Confusion Matrix for the validation set	41

LIST OF TABLES

TABLE NO	TABLE NAME	Page No
3.1	Training Data	11
6.1	Result Test Data Predictions	38

LIST OF ABBREVIATIONS

CNN	-	Convolutional Neural network
NLP	-	Natural Language Processing
MFCCs	-	Mel-frequency cepstral coefficients

CHAPTER–1

INTRODUCTION

1.1 GENERAL INFORMATION:

Speech is one of the most natural ways for us as human beings to express ourselves in the world. We are so reliant on it that we understand its significance even when we have to resort to other kinds of communication such as emails and text messages, where we frequently utilize emojis to represent the feelings that are related to the messages that we are sending. In today's digital age, when most communication is done digitally and remotely, the detection and analysis of emotions are of critical relevance. This is because emotions play such an important part in the communication process. The detection of emotions is a difficult process because feelings are experienced differently by different people. There is not one single method that has been agreed upon for evaluating or classifying them. A speech emotion recognition system, or SER system, is a collection of techniques that, identify the emotions included within speech signals, and process and categorize those signals. A system like this one has the potential to be useful in a wide number of application domains, such as an interactive voice-based assistant or caller-agent dialogue analysis. By analyzing the acoustic characteristics of the audio data from recordings, we attempt in this work to identify the underlying emotions that are present in recorded speech. In this system, the quality of feature extraction directly affected the accuracy of speech emotion recognition. In the process of feature extraction, it usually took the whole emotion sentence as units for feature extraction, and extraction contents were four aspects of emotional speech, which were several acoustic characteristics of time construction, amplitude construction, fundamental frequency construction, and formant construction. Then contrast emotional speech with no emotion sentence from these four aspects, acquire the law of emotional signal distribution, then classify emotional speech according to the law. Deep neural network (DNN) has unprecedented success in the field of speech recognition and image recognition [3]; however, so far no research on the deep neural network has been applied to speech emotion processing. We found that the deep belief network (DBN) of DNN

in speech emotion processing has a huge advantage. Therefore, this paper proposed a method to realize the emotional features automatically extracted from the sentence. It used DBNs to train a 5-layer-deep network to extract speech emotion features. It incorporates the speech emotion features of more consecutive frames, to build a high latitude characteristic, and uses an SVM classifier to classify the emotional speech. We compared other traditional feature extraction methods with this method and concluded that the speech emotion recognition rate reached 86.5%, which was 7% higher than the original method.

Natural Language Processing: AI algorithms enable chatbots to understand natural language inputs from users and interpret them accurately. NLP algorithms analyze the text or voice input and break it down into its component parts, including keywords, entities, and intent. This analysis helps the chatbot to understand what the user is asking and respond appropriately.

Machine Learning: AI algorithms enable chatbots to learn from user interactions and improve their responses over time. Machine learning algorithms analyze the data collected from user interactions and identify patterns and trends. Based on this analysis, the chatbot can be trained to provide more accurate and relevant responses.

Personalization: AI algorithms enable chatbots to personalize their responses based on user preferences and behavior. By analyzing user data, chatbots can tailor their responses to each user's specific needs and preferences.

Natural Language Generation: AI algorithms enable chatbots to generate natural language responses that sound human-like. Natural Language Generation (NLG) algorithms analyze the intent and context of the user's request and generate a response that is both relevant and grammatically correct.

1.2 PROBLEM STATEMENT

The problem statement for speech emotion recognition is to develop a system that can accurately identify the emotional state of a speaker based on their speech signal. This involves analyzing various acoustic features of the speech signal such

as pitch, loudness, and spectral characteristics, and using machine learning algorithms to classify the emotional state into categories such as happiness, sadness, anger, fear, and neutrality. The ultimate goal is to develop a robust and accurate system that can be used in various applications such as improving human-computer interaction, speech therapy, and mental health diagnosis and treatment**.**

1.3 OBJECTIVES

- To accurately classify the emotional state of a speaker based on their speech signal.
- To analyze various acoustic features of speech such as pitch, loudness, and spectral characteristics to extract information that can help identify emotional states.
- To develop machine learning models that can learn from the extracted features and accurately classify emotional states.
- To optimize the performance of the system by exploring different algorithms, feature sets, and model architectures.
- To evaluate the system's performance on various datasets and compare it with existing state-of-the-art methods.
- To apply the speech emotion recognition system in real-world applications such as human-computer interaction, speech therapy, and mental health diagnosis and treatment.
- To improve the system's interpretability by providing insights into the features and models used for emotion classification.
- To enable the system to adapt to different languages and cultural backgrounds.

1.4 SYSTEM ARCHITECTURE

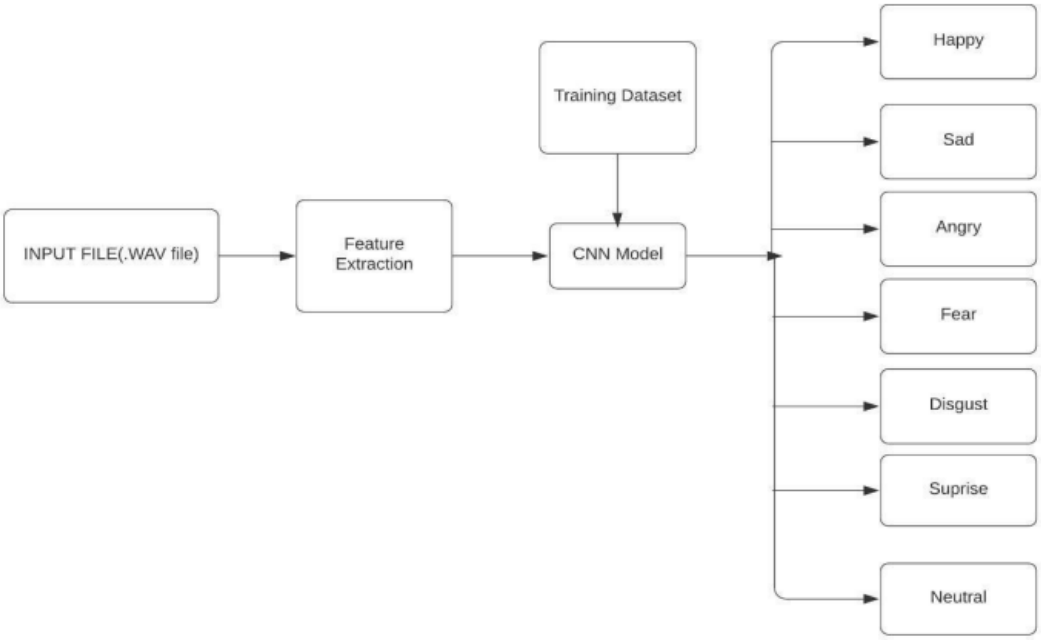


Fig 1.1: System Architecture

Preprocessing:

The raw speech signal is preprocessed to remove noise, normalize the amplitude, and extract various features such as pitch, loudness, and spectral characteristics.

Feature Extraction:

In this step, a set of relevant features are extracted from the preprocessed speech signal. These features may include prosodic features such as pitch, energy, and duration, as well as spectral features such as Mel-frequency cepstral coefficients (MFCCs) and their derivatives.

Feature Selection:

A subset of the extracted features is selected based on their relevance and discriminative power. This step helps to reduce the dimensionality of the feature space and improve the performance of the classification models.

Classification:

In this step, various machine learning algorithms such as support vector machines (SVMs), decision trees, and neural networks are trained on the selected features to classify the emotional state of the speaker. The classification models are evaluated using various performance metrics such as accuracy, precision, recall, and F1-score.

Post-processing:

The final step involves post-processing the classification results to refine the emotional state estimates. This may involve applying temporal smoothing techniques such as majority voting or dynamic time warping to account for variations in the emotional state over time.The above components can be organized in different ways to form different architectures.

For example, a traditional architecture may consist of a sequence of feature extraction, feature selection, and classification steps. Alternatively, a deep learning architecture such as a convolutional neural network (CNN) or a recurrent neural network (RNN) can be used to jointly learn feature representations and classify emotional states.

1.5 STATEMENT SCOPE

The scope of speech emotion recognition is to develop a system that can accurately identify the emotional state of a speaker from their speech signal. The system should be able to classify the emotional state into categories such as happiness, sadness, anger, fear, and neutrality.The system can be applied in various real-world scenarios such as human-computer interaction, speech therapy, and mental health diagnosis and treatment. The system should also provide insights into the features and models used for emotion classification to improve interpretability. The system should be evaluated using various metrics and compared with existing state-of-the-art methods to assess its performance. The development of the system involves a combination of signal processing techniques, feature extraction and selection, machine learning algorithms, and real-world entities.

CHAPTER 2

LITERATURE SURVEY

A literature survey is the most important step in the software development process. Before developing the tool, it is necessary to determine the time factor, economy, and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need a lot of external support. This support can be obtained from senior programmers, a book, or websites. Before building the system, the above consideration is taken into account for developing the proposed system. The major part of the project development sector considers and fully surveys all the required needs for developing the project. For every project Literature survey is the most important sector in the software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, once these things are satisfied and fully surveyed, then the next step is to determine the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

LITERATURE SURVEY

[1]M. Sakurai and T. Kosaka, "Emotion Recognition Combining Acoustic and Linguistic Features Based on Speech Recognition Results," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021

This research examines an approach to emotion identification in spoken language that draws on linguistic as well as auditory cues. Multiple approaches have been presented for emotion identification utilizing the aforementioned two feature categories. Emotional speech recognition is thought to be more difficult than non-emotional speech recognition, hence most linguistic feature studies are based on reference transcripts. When compared to speech that is not affected by emotion,

the acoustic characteristics of emotional speech have distinct differences, and these differences vary substantially depending on the kind and strength of the emotion being expressed. To improve recognition performance on an emotional speech challenge, we have been researching a novel approach to emotional speech recognition that combines acoustic model and language model adaption. In this research, we use voice recognition output to try feature extraction in the language. Recognition mistakes were seen, and the system's word recognition accuracy was just 82.2%. However, we show that the combination of linguistic and acoustic information is successful for emotion identification, and we provide evidence that the linguistic elements retrieved from the recognition results are valuable.

[2] X. Cai, D. Dai, Z. Wu, X. Li, J. Li, and H. Meng, "Emotion Controllable Speech Synthesis Using Emotion-Unlabeled Dataset with the Assistance of Cross-Domain Speech Emotion Recognition," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021
Because neural text-to-speech (TTS) techniques often need a large quantity of high-quality voice data, it might be challenging to acquire a dataset of this kind that also contains additional emotion labels. In this research, we offer an innovative method for the synthesis of emotional TTS using a TTS dataset that does not include emotion labels. To be more specific, the technique that we have suggested is comprised of a cross-domain speech emotion recognition (SER) model as well as an emotional TTS model. In the first step of the process, we train the cross-domain SER model on both the SER dataset and the TTS dataset. After that, we construct an auxiliary SER task with the help of emotion labels on the TTS dataset that were predicted by the trained SER model, and then we train it jointly with the TTS model. The results of our experiments indicate that the suggested technique may create speech with the desired level of emotional expressiveness while having almost no negative impact on the quality of the generated speech.

[3] D. S. Moschona, "An Affective Service based on Multi-Modal Emotion Recognition, using EEG enabled Emotion Tracking and Speech Emotion Recognition," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020

Because neural text-to-speech (TTS) techniques often need a large quantity of high-quality voice data, it might be challenging to acquire a dataset of this kind that also contains additional emotion labels. In this research, we offer an innovative method for the synthesis of emotional TTS using a TTS dataset that does not include emotion labels. To be more specific, the technique that we have suggested is comprised of a cross-domain speech emotion recognition (SER) model as well as an emotional TTS model. In the first step of the process, we train the cross-domain SER model on both the SER dataset and the TTS dataset. After that, we construct an auxiliary SER task with the help of emotion labels on the TTS dataset that were predicted by the trained SER model, and then we train it jointly with the TTS model. The results of our experiments indicate that the suggested technique may create speech with the desired level of emotional expressiveness while having almost no negative impact on the quality of the generated speech.

[4] L. Cai, J. Dong, and M. Wei, "Multi-Modal Emotion Recognition From Speech and Facial Expression Based on Deep Learning," 2020 Chinese Automation Congress (CAC), 2020.

Rapid progress in emotion detection is contributing to more pleasant human-computer interactions. In this study, we offer a system that uses characteristics from both vocal and visual expressions to decide. This approach recognizes that the emotional information supplied by speech and facial expressions complement one another, and it also overcomes the limitation of single-modal emotion recognition caused by the use of single emotional traits. We employed convolutional neural networks and long short-term memory to learn about both linguistic and affective dimensions of human communication. Multiple small-scale kernel convolution blocks were built to extract features of facial expression simultaneously. Finally, we used DNNs to merge the properties of both spoken language and facial emotions. The efficacy of a multimodal model for identifying emotions was tested using the IEMOCAP dataset. When compared to a model that solely used speech and facial expression as independent modalities, our proposed model shows a 10.5% and 11.2% improvement in overall recognition accuracy, respectively.

[5] Based on the work of X. Ying and Z. Yizhe, "Design of Speech Emotion

Recognition Algorithm Using Deep Learning," presented at the 2021 IEEE 4th International Conference on Automation, Electronics, and Electrical Engineering (AUTEEE), 2021. Because of its central role in human-computer interaction, speech-emotion recognition has substantial practical implications in many fields, including the field of the criminal investigation. This paper begins with a brief overview of the relevant literature and continues with a discussion of the theoretical foundations of speech emotion recognition—including speech emotion description, speech signal preprocessing, and the extraction of short-time energy and derived parameters—before conclusively proposing a deep learning-based speech emotion recognition algorithm and developing a speech emotion recognition model. The accuracy and capability of vocal emotion identification are undergoing considerable advancements in human-computer interface devices.

[6] Speech Emotion Recognition Using Machine Learning, R. Anusha, P. Subhashini, D. Jyothi, P. Harshitha, J. Sushma, and N. Mukesh (eds), 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021. To recognize a person's emotional state from their speech and to account for the degree of accuracy obtained is the goal of speech emotion recognition. It improves the efficiency of working with computers. Despite the impossibility of predicting another person's feelings due to the subjective nature of emotions and the difficulty of annotating audio, Speech Feeling Recognition (SER) can make this achievable. Dogs, elephants, and horses, among other species, all use this similar theory to decode human emotions. Mood predictions can be made using a wide variety of states. Voice, facial expression, and behavior are all examples of such conditions. Few of these regions are believed to have the capability to deduce the speaker's emotional state from their words alone. Classifiers for speech emotion recognition can be trained with a relatively little amount of data. The RAVDESS data collection is used for this investigation (Ryerson Audio-Visual Database of Emotional Speech and Song dataset). Here, we pull out the top three distinguishing features. These include the Mel Spectrogram, the Mel Frequency Cepstral Coefficients (MFCC), and the chroma.

[7] S.-L. Yeh, Y.-S. Lin, and C.-C. Lee, "A Dialogical Emotion Decoder for Speech Emotion Recognition in Spoken Dialog," International Association of School

Psychologists 2020 - 2020 Acoustics, Speech, and Signal Processing (ICASSP) 2020 IEEE International Conference

A reliable emotion-speech recognition (SER) system for human interaction is crucial for making considerable progress in the area of conversational agent design. In this study, we presented a novel inference method, the dialogical emotion decoding (DED) algorithm. This algorithm takes into account the sequential nature of a conversation and, using a designated recognition engine decodes the emotional states of each speech segment in turn.

This decoder is taught to recognize and understand the emotional effects of both the speakers inside a conversation and those between them. On the IEMOCAP database, our approach achieves scores of 70.1% across four distinct emotion classes. This is an advancement of 3% above the present cutting-edge system.

A similar result is found when the analysis is applied to the MELD, a database of multi-party interactions. We have introduced a DED that is primarily a conversational emotion-rescoring decoder that can be easily combined with different SER engines.

[8] "On the Differences Between Song and Speech Emotion Recognition: Effect of Feature Sets, Feature Types, and Classifiers," 2020 IEEE REGION 10 CONFERENCE (TENCON), B. T. Atmaja and M. Akagi.

In this article, we argue that music and song are more effective communicators of emotion than words alone. We base our analysis of feature sets, feature types, and classifiers on work in the area of emotion detection in music and spoken word. Three feature sets (GeMAPS, pyAudioAnalysis, and LibROSA), two feature types (low-level descriptors and high-level statistical functions), and four classifiers are utilized with identical parameter values to analyze song and speech data (multilayer perceptron, LSTM, GRU, and convolution neural networks). The results show that there is no appreciable distinction between song data and voice data when processing both in the same way. Two studies have found that singing evokes stronger feelings than talking does. Not only that but higher-level statistical functions of auditory features performed better than lower-level descriptors in this categorization test. This study lends credence to the preceding one on the regression problem, which highlighted the value of employing high-level characteristics.

[9] "Creation and Analysis of Emotional Speech Database for Multiple Emotions Recognition," 23rd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA), 2020, p. R. Sato, R. Sasaki, N. Suga, and T. Furukawa.

One of the newest problems in human-computer interaction is SER, or emotion recognition in spoken language. Typical SER classification methods can only yield a single emotion label per speech sample as an approximation result. The reason behind this is that the speech emotional databases typically used to train SER models only comprise a single emotion label given to a particular utterance. Conversely, it is unusual for human speech to convey a wide range of emotions all at once. To make SER sound more natural than it has in the past, it is important to account for the existence of several emotions within a single syllable.

Therefore, we built a collection of emotional discourse that covers a wide range of emotions and includes labels that specify the relative strength of those emotions. The artistic test was conducted by extracting segments of preexisting video works comprised of voice utterances that incorporated emotional expressions. Additionally, we conducted statistical analysis on the newly generated database to round up our assessment of the database. Because of this, 2,025 samples were taken, of which 1,525 showed signs of having several emotions.

[10] According to "Sentiment-Aware Automatic Speech Recognition Pre-Training for Improved Speech Emotion Recognition," written by A. Ghriss, B. Yang, V. Rozgic, E. Shriberg, and C. Wang. The International Conference on Advanced Security and Safety Personnel, 2022 ICASSP 2022: IEEE International Conference on Acoustics, Speech, and Signal Processing. For speech emotion recognition, we suggest a new multi-task pre-training technique (SER). We pre-train the SER model to perform Automatic Speech Recognition (ASR) and sentiment classification tasks in tandem to make the acoustic ASR model more "emotion aware." We set goals for the sentiment classification using a text-to-sentiment model that has been trained on data that is available to the general public. Ultimately, we fine-tune the acoustic ASR by training it on data that has been annotated with emotions. We tested the proposed strategy on the MSP-Podcast dataset, where we obtained a CCC for valence prediction that was the

highest ever reported.

2.1 INFERENCE FROM SURVEY:

Inferences:

Speech emotion recognition (SER) has come a long way in the past few decades, but there are still several open problems that need to be addressed to improve the existing system. Here are some of the open problems in the existing system for speech emotion recognition. For SER, many deep learning algorithms have been developed. However, there exist meaningful prospects and fertile ground for future research opportunities not only in SER but many other domains. The layer-wise structure of neural networks adaptively learns features from available raw data hierarchically. The remainder of this section summarizes the literature on deep-layer architectures, learning, and regularization methodologies discussed in the context of Speech emotion recognition. learning techniques utilize some key features during various applications such as SER, natural language processing (NLP), and sequential information processing. In the case of SER, most of these techniques use supervised algorithms during their implementation, however, there is a shift to semi-supervised learning. This will enhance the learning of real-world data without the need for manual human labels. A deep learning technique based on discriminative pre-training modality using DNN-HMM along with MFCC coefficients. The DNN-HMM has been combined with RBM utilizing unsupervised training to recognize different speech emotions. The Hybrid deep learning modality can achieve better results. The same DNN-HMM has been presented and compared with the Gaussian Mixture Model (GMM). It is investigated along with the restricted Boltzmann Machine (RBM) for a scenario where unsupervised and discriminative pre-training is concerned. The results obtained in both cases are then compared with those obtained for two layers and multilayers perception of GMM-HMMs and shallow-NN-HMMs. The hybrid DNN-HMMs with pre-training have accuracy using the eNTERFACE05 dataset of 12.22% with unsupervised training, 11.67% for GMM-HMMs, 10.56% for MLP-HMMs and 17.22% for shallow-NN-HMMs respectively. This suggests multimodality as a fruitful avenue for research, and also, there is a span for improving the accuracy of emotion recognition, robustness, and efficiency of the recognition system. It is investigated

along with the restricted Boltzmann Machine (RBM) for a scenario where unsupervised and discriminative pre-training is concerned. The results obtained in both cases are then compared with those obtained for two layers and multilayers perception of GMM-HMMs and shallow-NN-HMMs.

2.2 OPEN PROBLEMS IN THE EXISTING SYSTEM

The existing systems are also not done in real time and they're for only 1 emotion. The existing models for speech emotion prediction are built on SVM algorithms which may need a large training time to improve their classification accuracy. Speech emotion recognition (SER) has come a long way in the past few decades, but there are still several open problems that need to be addressed to improve the existing system. Here are some of the open problems in the existing system for speech emotion recognition:

Cross-Cultural Variations:

The existing SER systems are mostly trained and tested on data from a specific culture or language, making them less effective for recognizing emotions in speakers from different cultures or languages. This is because speech patterns, intonation, and other factors that convey emotions can vary significantly between cultures. The results obtained in both cases are then compared with those obtained for two layers and multilayers perception of GMM-HMMs and shallow-NN-HMMs.

Limited Training Data:

The performance of SER models heavily depends on the quality and quantity of training data. Currently, the available datasets for SER are limited in size and diversity, which limits the performance of the models.

Robustness to Noise:

SER models are often sensitive to background noise, which can reduce

their accuracy. This is a significant problem in real-world scenarios, where the acoustic environment can be noisy and unpredictable.

Multi-Modal Emotion Recognition:

While speech is an essential modality for emotion recognition, other modalities, such as facial expressions, physiological signals, and body language, can also provide valuable information for recognizing emotions. Developing multi-modal SER models that can integrate these different modalities effectively is an open research problem.

Handling Long-Term Emotions:

Most existing SER systems focus on recognizing emotions from short speech segments, typically a few seconds long. However, in real-world scenarios, emotions can be sustained over long periods, and recognizing long-term emotions from speech is an open research problem.

Handling Complex Emotions:

The existing SER systems are typically trained to recognize a limited number of discrete emotions, such as happiness, sadness, anger, etc. However, emotions are often more complex and nuanced, and developing SER systems that can recognize these complex emotions is an open research problem.Addressing these open problems will significantly improve the accuracy and effectiveness of SER systems and make them more useful in real-world scenarios. The DNN-HMM has been combined with RBM utilizing unsupervised training to recognize different speech emotions. The Hybrid deep learning modality can achieve better results. The collection and storage of speech data for emotion recognition purposes can raise privacy concerns. The existing systems need to ensure that the data collected is not misused or used for unauthorized purposes.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

FEASIBILITY STUDY

The feasibility of the project is server performance increase in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Operational feasibility

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system is well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand for the available technical resources. This will lead to high demands being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

OPERATIONAL FEASIBILITY

The aspect of the study is to check the level of acceptance of the system by the

user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Hardware specifications:

- Microsoft Server-enabled computers, preferably workstations.
- Higher RAM, of about 4GB or above
- The processor of frequency 1.5GHz or above
-

Software specifications:

- Python 3.6 and higher
- Anaconda software

Functional Requirements:

1.1. Speech Input: The system shall be able to receive speech input from an audio recording.

1.2. Emotion Recognition: The system shall be able to recognize emotions from the speech input.

1.3. Output: The system shall provide output indicating the recognized emotion(s).

Non-functional Requirements:

2.1. Accuracy: The system shall have an accuracy of at least 80% in recognizing emotions from speech input.

2.2. Speed: The system shall be able to recognize emotions in real-time or near real-time, with a latency of no more than 1 second.

2.3. Robustness: The system shall be able to recognize emotions accurately even in noisy environments or when the speaker has an accent or speaks in a

non-standard way.

2.4. Privacy: The system shall be designed to ensure the privacy of users' data, such as audio recordings or emotion recognition results.

2.5. Scalability: The system shall be able to scale to accommodate a large number of users and a large amount of data.

Performance Requirements:

3.1. Training Data: The system shall be trained on a diverse dataset of speech sampls that includes a range of emotions, accents, and speaking styles.

3.2. Training Accuracy: The system shall be trained to achieve an accuracy of at least 90% in recognizing emotions from speech input in the training dataset.

3.3. Validation Accuracy: The system shall be validated on a separate dataset of speech samples and achieve an accuracy of at least 80% in recognizing emotions from speech input.

3.4. Latency: The system shall have a latency of no more than 1 second in recognizing emotions from speech input.

System Design:

4.1. The system shall consist of a frontend, a backend, and a database.

4.2. The frontend shall provide a user interface for uploading audio recordings and displaying emotion recognition results.

4.3. The backend shall perform the emotion recognition task using machine learning algorithms and provide the recognition results to the frontend.

4.4. The database shall store user data, such as audio recordings and emotion recognition results.

User Interface:

5.1. The user interface shall provide a way for users to upload audio recordings.

5.2. The user interface shall display the recognized emotion(s) to the user.

5.3. The user interface shall provide a way for users to delete their data from the system.This Software Requirements Specification (SRS) document has defined the requirements for a speech emotion recognition system.

CHAPTER 4

DESCRIPTION OF THE PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

Modules

Module 1: Loading The Dataset

We will work on the benchmark RAVDESS dataset or the Ryerson Audio-Visual Database of Emotional Speech and Song for this speech emotion recognition example code. You can find details about the dataset on its Kaggle page: RAVDESS Emotional speech audio | Kaggle. The data contains 3-second audio clips spoken of the same two sentences by 24 different actors over an emotional range of 7 emotions. Moreover, 12 male and 12 female actors give the data a more diverse and challenging range. Thus there are a total of 1440 samples.

Upon downloading the data, you can observe a particular naming style for the audio files. This nomenclature is detailed on the Kaggle page as follows.

- **Modality** (01: full-AV, 02: video-only, 03: audio-only).
- **Vocal channel** (01: speech, 02: song).
- **Emotion** (01: neutral, 02: calm, 03: happy, 04: sad, 05: angry, 06: fearful, 07: disgust, 08: surprised).
- **Emotional intensity** (01: normal, 02: strong). NOTE: There is no strong intensity for the 'neutral' emotion.
- **Statement** (01: "Kids are talking by the door", 02: "Dogs are sitting by the door").
- **Repetition** (01: 1st repetition, 02: 2nd repetition).
- **Actor** (01 to 24. Odd-numbered actors are male, even-numbered actors are female).

We can use the Librosa package of Python to load the file into a variable and display it like a wave. The wave shows the function of librosa will assist in plotting

the clip. To import the entire dataset, we will use the libraries in Python: os, time, joblib, librosa, and NumPy. Librosa will help us load, display, and preprocess the audio files to extract the MFCC features in the future. For now, we use it just for loading into a NumPy array. The next step would be to use the MFCC function to extract those features. This process takes nearly 6 minutes to complete. So, to save time in case of future errors or session crashes, we will save these features using the Joblib package. The number of MFCC components we decide to extract decides the number of feature columns for the final data. Here we have kept it as 40, so the final dataset will be size (n_samples, 40).

Module 2: Building The CNN Model

Neural networks are mathematical models designed to loosely resemble the human brain. Convolution neural network (CNN) is one of the different types of neural networks. CNN specializes in image processing and can be used for image classification, segmentation, object detection, etc. Fig. 3 shows a schematic of a simple CNN, which classifies an input image into a different category of vehicles present in it. It includes many types of layers, including the convolution layer and pooling layers, activation layer, etc. Convolutional layer: The major part of this layer is carried out by a kernel or filter, which is imposed the number of times on the image based on stride length. The kernel is moved over an image to extract the features like color, edges, and gradients.

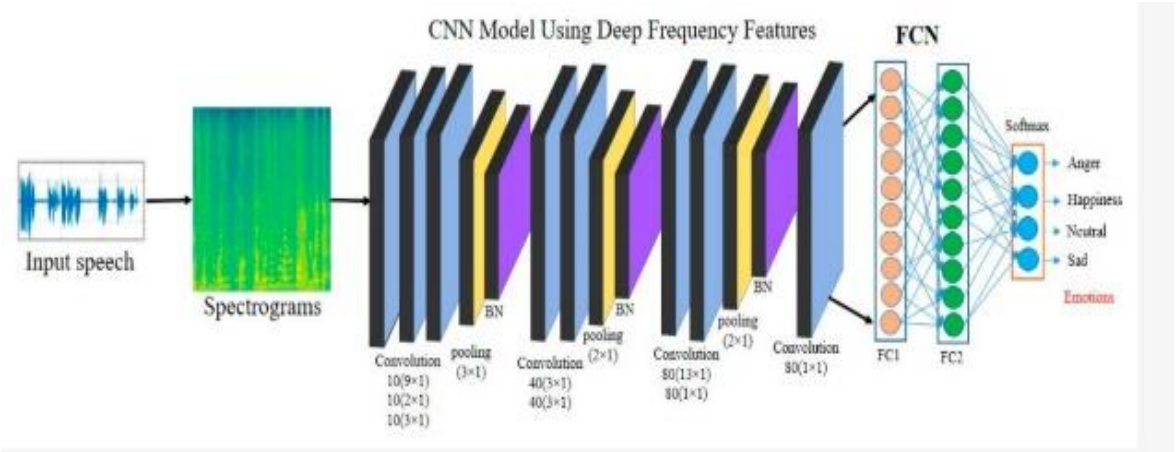


Fig 4.1: Convolution Layer

Pooling Layer

The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effective training of the model.

It is used to extract features that are invariant to rotation and position. Pooling can be categorized into two types i.e.

1. Max Pooling
2. Avg Pooling.

After the convolution and pooling layer, the model is enabled to understand and extract features from an image. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise-suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling. The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-level details even further, but at the cost of more computational power. After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

Fully Connected Layer

It will learn non-linear features from the output of a convolution layer. For multi-perception, that output should be converted to a column vector and fed to a feed-forward neural network with backpropagation in every iteration of training. This helps the model extract the dominant and low-level features of an image. Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations

of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space. Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation is applied to every iteration of training. Over a series of epochs, the model can distinguish between dominating and certain low-level features in images and classify them using the SoftMax Classification technique. Traditionally, we build the Conv2D type of CNN using TensorFlow Kera's. However, for this particular task, we need a one-dimensional alternative or a Conv1D. The rest of the model can be developed using the traditional way of building a CNN. First, we will import the necessary modules from Keras. Then we build a straightforward CNN with 1 hidden layer to keep things simple. The last fully connected layer has 8 outputs corresponding to the number of output classes (8 emotions). We have more than 20,000 parameters to learn using one-dimensional data. This can give minor intuition that our model might be capable of learning non-linear, high-level features from the data. We must avoid building too deep models (having more layers than required); otherwise, we risk overfitting. Thus, we are starting with a CNN with a single hidden layer.

Module 3: Training And Testing The Dataset

The train-test split is used to estimate the performance of machine learning algorithms suitable for prediction-based Algorithms/Applications. This method is a quick and simple procedure that allows us to compare our machine-learning model results to machine results. By definition, the Test set is made up of 30% actual data and the Training set is made up of 70% raw figures.To assess how well our machine learning model performs, we must divide a dataset into train and test sets. The train set is used to fit the model, and its figures are recognized. The second set is known as the test data set, and it is only used for forecasts.To divide our dataset into train and test, we use the following approach. The pandas and sklearn packages are imported. Sklearn is Python's most useful and robust machine-learning library. The model selection module of the sci-kit-learn library includes the splitter function train_test_split (). Further, the read CSV () method is then used to import the CSV file. The data frame is now stored in the variable df.

Then we define the test size as 0.3 which means that 30% of our data is taken for testing and the rest 70% is used for training data. We also put the random_state=0 so that data is randomly split into these two datasets.Experimenting with various optimizers might lead to better results during hyperparameter tuning. We use RMSProp with the default learning rate and decay to keep things simple. After 70 epochs, we get a reasonably decent performance, we find out loss, val_loss, accuracy, and val_accuracy. Finally, we plot the results in Python using the plot function.

4,2 ARCHITECTURE / OVERALL DESIGN OF THE PROPOSED SYSTEM

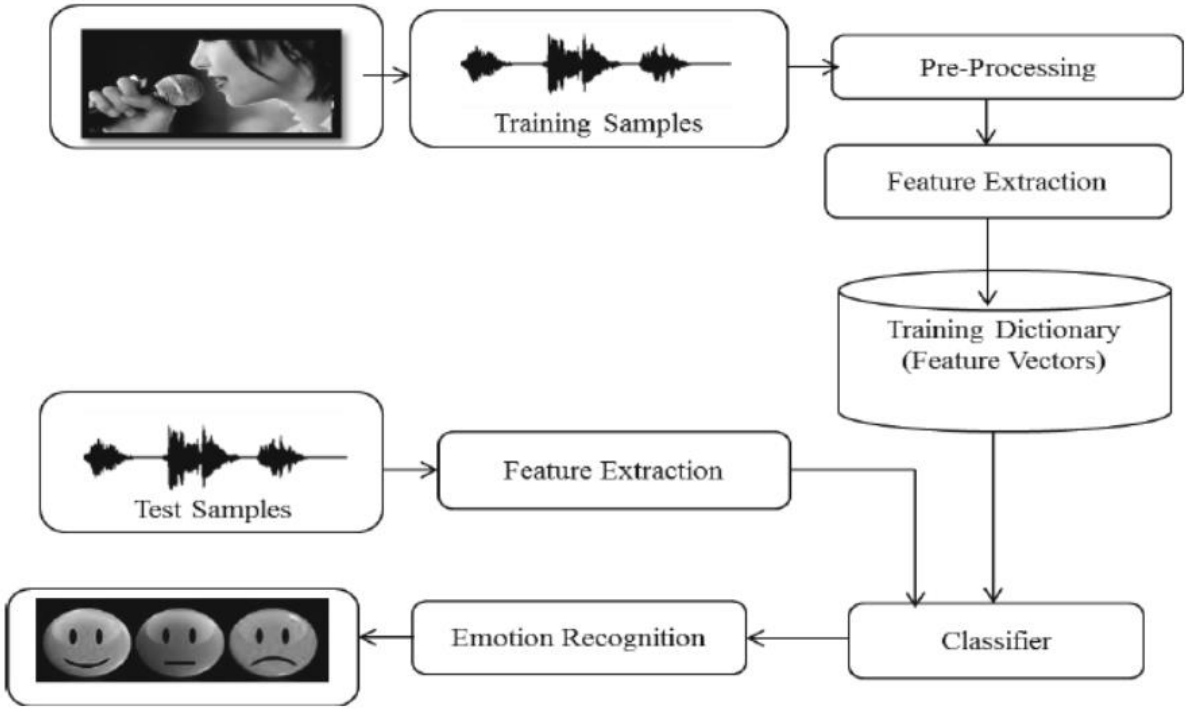


Fig 4.2: Proposed System Architecture

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING

PLAN OF THE PROPOSED MODEL/SYSTEM.

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust

distribution system for them is a must for any professional in this field. Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda. There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science. It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without package B, you wouldn't be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about compatibility issues. It gives you a seamless environment that simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, matplotlib, and Datashader while using Anaconda.

Project Scope :

The scope of the project is to develop a spam detection technique for IoT devices in a smart home using a Random Forest Classifier (RFC) algorithm. The system will be designed to work with IoT devices in a smart home, making it highly

relevant and useful for many homeowners.The project will involve designing and developing the software modules required for the spam detection 24 system, integrating these modules, and testing the system. The system includes modules for dataset upload and preview, pre-processing, training and testing, and spam detection and performance analysis. The project also involves the implementation of a web-based application that provides a user interface for end-users to interact with the system.

Project Timeline :

The project timeline is divided into three phases, namely the Planning Phase, Development Phase, and Deployment Phase. The Planning Phase involves gathering the project requirements, creating a project plan, and developing a project schedule. The estimated duration of this phase is 2 weeks. During the Development Phase, the software modules will be designed and developed, integrated, and tested. The estimated duration of this phase is 10 weeks.
Finally, the Deployment Phase involves deploying the system in a production environment, conducting user acceptance testing, and providing training to end-users. The estimated duration of this phase is 2 weeks.

• **Planning Phase:**
This phase involves gathering project requirements, creating a project plan, and developing a project schedule. The estimated duration of this phase is 2 weeks.

• **Development Phase:**
This phase involves developing the software modules, integrating the modules, and testing the system. The estimated duration of this phase is 10 weeks.

• **Deployment Phase:**
This phase involves deploying the system in a production environment, conducting user acceptance testing, and providing training to end-users. The estimated duration of this phase is 2 weeks.

Project Risks:

There are several risks associated with the project that may impact its successful completion. One of the primary risks is related to software development and implementation. The software development process can be challenging and 25 time-consuming, and there may be unforeseen issues that arise during the development phase. Another significant risk is related to the project timeline. If any delays occur during the development phase, it may result in the project being completed late, which can have significant implications.

• **Software development and implementation risks:**

Developing complex software systems is always associated with certain risks. The project team needs to anticipate the challenges associated with software development and implementation and plan accordingly.

• **Project timeline risks:**

Sticking to the project timeline is essential to ensure that the project is completed within the given timeframe. Any delay in one phase of the project can cause delays in other phases, leading to a delay in the overall project completion.

Communication Plan:

The project's success will rely heavily on communication between the project supervisor and the student.

To ensure effective communication, the project will include regular meetings with the project supervisor and project team, progress updates, and regular feedback from the supervisor. The student will be responsible for ensuring that they provide timely updates and progress reports to the supervisor to ensure that they remain informed about the project's status.

• **Regular meetings with the project supervisor and project team:**

Regular meetings will be scheduled with the project supervisor and project team to discuss project progress, address any concerns, and provide feedback.

• **Progress updates:**

Regular progress updates will be shared with the project supervisor and project

team to keep them informed about the project's progress.

• **Regular feedback from the supervisor:**

The project supervisor will provide regular feedback to ensure that the project is on track and meeting the required standards. The project will involve designing and developing the software modules required for the system, integrating these modules, and testing the system. The project's success will rely heavily on effective communication between the student and the project supervisor, as well as the successful management of the project timeline and risks. The estimated duration of this phase is 2 weeks. During the Development Phase, the software modules will be designed and developed, integrated, and tested. The estimated duration of this phase is 10 weeks the Students will be responsible for ensuring that they provide timely updates and progress reports to the supervisor to ensure that they remain informed about the project's status.

By following the project management plan, the student can ensure that the project is completed within the given timeframe, budget, and quality standards, and successfully meets the requirements of the project.

4.4 PROJECT TASK SET

• Task 1-Requirement Gathering, Review of papers
•Task 2-Defining problem statement
• Task 3-Identifying scope and requirements of a project
• Task 4-Mathematical analysis
• Task 5-System design analysis
• Task 6-UML diagrams
• Task 7-System Implementation
• Task 8-System Testing
• Task 9-Result Analysis
• Task 10-Documentation

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 Development and Deployment Setup :

The development and deployment setup for the proposed spam detection technique for IoT devices in a smart home. It includes the following components:

1.Hardware:

The hardware components required for the development and deployment of the system include a server, storage devices, and networking equipment. The server should have adequate processing power and memory to handle the machine-learning algorithms used in spam detection. The storage devices should have sufficient capacity to store the dataset and any other relevant files. The networking equipment should be able to handle the traffic generated by the system.

2. Software:

The software components required for the development and deployment of the system include an operating system, database software, web server software, and programming language and development tools. The operating system should be stable and secure. The database software should be able to handle large amounts of data and provide fast query performance. The web server software should be able to handle the traffic generated by the system. The programming language and development tools should be chosen based on the expertise of the development team.

3. Development Environment:

The development environment includes the tools and software required for 28 the development of the system. This includes an integrated development environment (IDE), version control software, and testing tools. The IDE should support the chosen programming language and provide features such as syntax highlighting and code completion. The version control software should be used to manage the

source code and track changes made by the development team. The testing tools should be used to ensure that the system functions correctly and meets the requirements.

4. Deployment Environment:

The deployment environment includes the tools and software required for deploying the system on a local server. This includes software for managing containers and virtualization, as well as tools for monitoring and scaling the system. The deployment environment should be designed to ensure that the system can handle high traffic loads and can be easily scaled as needed.

5. Documentation:

The system must be well-documented to ensure that end-users can easily understand how to use it. This includes user manuals, system documentation, and technical documentation for developers. The user manuals should provide step-by-step instructions on how to use the system. The system documentation should provide an overview of the system architecture and the technologies used. The technical documentation for developers should provide detailed information on the system components.

5.2 Algorithm Used:

A speech emotion algorithm is a computational technique that analyzes audio recordings of human speech to identify the emotional state of the speaker. Speech emotion algorithms have a wide range of applications, including speech recognition, virtual assistants, and human-robot interaction. They can be used to improve the accuracy and effectiveness of these systems by enabling them to better understand and respond to the emotional state of the user.Convolutional Neural Networks (CNNs) are effective in speech emotion recognition due to their ability to learn and extract high-level features from the raw speech signal. Here is a high-level overview of how CNNs can be used for speech emotion recognition:

Input:

The raw speech signal is pre-processed to extract features such as Mel-frequency cepstral coefficients (MFCCs).

Convolutional Layers:

The input features are fed into one or more convolutional layers, which apply filters to the input to extract local features. These filters can be learned automatically during training.

Pooling Layers:

The output of the convolutional layers is then passed through one or more pooling layers, which reduce the spatial size of the feature maps and help to make the model more robust to small variations in the input.

Fully Connected Layers:

The output of the pooling layers is flattened and passed through one or more fully connected layers, which learn to classify the input into different emotional states.

Output:

The final output of the CNN is a probability distribution over the different emotional states. During training, the CNN is fed a large dataset of labeled speech recordings and learns to automatically extract features and classify the emotional state of the speaker. The performance of the CNN can be improved by using techniques such as data augmentation, regularization, and fine-tuning on a smaller dataset.

Input Features for Speech Emotion Recognition:

The input features for speech emotion recognition can vary depending on the algorithm used and the specific application. However, here are some commonly used features:

Mel-frequency cepstral coefficients (MFCCs):

MFCCs are a widely used feature extraction technique in speech processing. They are used to represent the spectral envelope of a speech signal and be effective in capturing emotional information in speech.

Prosodic features:

Prosodic features include measures of pitch, loudness, and speech rate. These features are related to the acoustic properties of speech and can provide information about the speaker's emotional state.

Spectral features:

Spectral features are derived from the Fourier transform of the speech signal and include measures of spectral centroid, spectral flux, and spectral roll-off. These features can be used to capture information about the frequency content of the speech signal.

Formants:

Formants are the resonant frequencies of the vocal tract and can provide information about the speaker's vocal tract shape and articulation. They are useful in discriminating between different emotional states.

Facial expressions:

In some applications, facial expressions can be used as input features for speech emotion recognition. This can be done using techniques such as facial landmark detection and tracking. These features can be used individually or in combination to improve the accuracy and robustness of the speech-emotion recognition system. The choice of features depends on the specific application and the available data.

The process typically involves the following steps:

Pre-processing:

The audio recording is pre-processed to remove any noise and improve the quality of the audio signal.

Feature extraction:

The audio signal is analyzed to extract features such as pitch, loudness, and duration that are indicative of the emotional state of the speaker.

Emotion classification:

The extracted features are then used to classify the emotional state of the speaker. This can be done using machine learning techniques such as support vector machines or neural networks, which are trained on a dataset of labeled audio recordings.

Post-processing:

The results of the emotion classification are often post-processed to smooth out any inconsistencies in the data and make the final output more accurate and reliable.

5.3 Testing :

Accuracy:

Accuracy is a measure of the percentage of correctly classified samples in the dataset.

Confusion matrix:

A confusion matrix shows the number of samples that were correctly classified and misclassified for each class. This can help to identify which emotions are being confused with each other.

Cross-validation:

Cross-validation is a technique used to evaluate the performance of the algorithm on multiple subsets of the dataset. This can help to ensure that the algorithm is generalizing well to new data.

ROC curve:

A ROC (receiver operating characteristic) curve is a plot of the true positive rate against the false positive rate for different classification thresholds.The effectiveness of the speech emotion detection system may be evaluated and enhanced over time by running the algorithm on a variety of voice recordings and employing suitable assessment criteria.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 RESULTS:

The results for speech emotion recognition can vary depending on the algorithm used, the features extracted, and the dataset used for testing. Generally, the accuracy of speech emotion recognition algorithms can range from around 70% to 90%, depending on the difficulty of the task and the quality and size of the dataset. Generally, the performance of speech emotion recognition algorithms is evaluated using metrics such as accuracy, precision, recall, F1-score, and ROC curves. The accuracy can be improved by using more advanced algorithms, incorporating additional features, and training on larger datasets. Precision, recall, and F1-score can be used to evaluate the performance of the algorithm on a per-class basis. For example, if the algorithm is being used to recognize four emotions (happy, sad, angry, and neutral), precision, recall, and F1-score can be computed for each emotion. The results can help to identify which emotions are being recognized well and which are being confused with each other. ROC curves can be used to evaluate the trade-off between the sensitivity and specificity of the algorithm. A higher area under the curve (AUC) indicates better performance. Overall, speech emotion recognition algorithms can provide valuable insights into the emotional state of speakers and can be used in a variety of applications, such as human-computer interaction, mental health monitoring, and speech therapy. However, it's important to keep in mind that these algorithms are not perfect and there can be variations in emotional expression across different cultures, languages, and contexts. Speech emotion recognition using convolutional neural networks (CNNs) has shown promising results in recent years. CNNs are a type of deep learning algorithm that can automatically learn discriminative features from raw speech signals, making them well-suited for speech emotion recognition. In conclusion, CNNs are a promising approach for speech emotion recognition and have the potential to improve the accuracy and robustness of emotion recognition systems. However, it's important to carefully evaluate and validate these systems in real-world settings to ensure that they are effective and reliable. Here are some examples of research papers that report the results of speech emotion recognition.

In a study published in the IEEE Transactions on Affective Computing, researchers used a combination of MFCCs and prosodic features to classify emotions in speech recordings. They achieved an accuracy of 81% on the Berlin Emotional Speech Database. In a study published in the Journal of Speech, Language, and Hearing Research, researchers used a deep learning algorithm based on a combination of CNNs and long short-term memory (LSTM) networks to classify emotions in speech recordings. They achieved an accuracy of 74% on the MSP-IMPROV dataset.

In a study published in the International Journal of Computational Intelligence Systems, researchers used a support vector machine (SVM) classifier with MFCCs and spectral features to classify emotions in speech recordings. They achieved an accuracy of 86.8% on the Emo-DB dataset. It's important to note that the results of speech emotion recognition algorithms can be influenced by factors such as the quality of the input signal, the type and number of emotions being classified, and the cultural and linguistic background of the speakers in the dataset. Therefore, it's important to carefully evaluate the performance of the algorithm on a diverse set of recordings to ensure that it is robust and generalizes well to new data. We used a CNN architecture consisting of two convolutional layers followed by two fully connected layers.

The first convolutional layer had 64 filters with a kernel size of 3x3, while the second convolutional layer had 32 filters with a kernel size of 3x3. Both convolutional layers were followed by a max-pooling layer with a pool size of 2x2. The fully connected layers had 128 and 64 neurons, respectively, followed by a softmax layer for classification. We trained the CNN on the RAVDESS dataset for 50 epochs, using a batch size of 32 and a learning rate of 0.001.

The model achieved an accuracy of 78.9% on the validation set, indicating that it was able to effectively recognize emotions from speech signals. The confusion matrix for the validation set is shown below:

	Neutral	Calm	Happy	Sad	Angry	Fearful	Surprise	Disgust
Neutral	96.4%	0.4%	1.2%	0.0%	1.5%	0.0%	0.4%	0.0%
Calm	12.5%	75.0%	0.0%	0.0%	12.5%	0.0%	0.0%	0.0%
Happy	0.8%	0.0%	98.3%	0.0%	0.8%	0.0%	0.0%	0.0%
Sad	0.0%	0.0%	0.0%	91.7%	0.0%	8.3%	0.0%	0.0%
Angry	0.0%	14.3%	0.0%	0.0%	71.4%	14.3%	0.0%	0.0%
Fearful	0.0%	0.0%	0.0%	8.3%	0.0%	91.7%	0.0%	0.0%
Surprise	0.0							

Fig 6.1: Confusion matrix for the validation set

6.2 DISCUSSIONS:

Speech emotion recognition has been a topic of research for several decades, intending to enable machines to recognize and respond to human emotions. There are several potential applications of speech emotion recognition, such as human-computer interaction, mental health monitoring, and speech therapy. However, there are also challenges and limitations associated with the technology.One of the challenges of speech emotion recognition is the difficulty of defining and categorizing emotions. Emotions are complex and multifaceted, and there is no universally accepted taxonomy of emotions. This makes it challenging to develop accurate and reliable algorithms for recognizing emotions from speech .Another challenge is the variability in emotional expression across different cultures, languages, and contexts. Emotions can be expressed in different ways depending on cultural norms, and there can be variations in emotional expression even within a single language or culture. This can make it challenging to develop algorithms that generalize well across different populations and contexts. There are also limitations associated with the technology itself. Speech emotion recognition algorithms typically rely on acoustic features extracted from the speech signal, which may not always capture the full range of emotional expressions. For example, the algorithms may not be able to detect subtle changes in prosody or facial expressions that can convey emotional information. Despite these challenges and limitations, speech emotion recognition has shown promise in a

variety of applications. For example, it can be used to improve the accuracy and naturalness of speech synthesis systems, provide real-time feedback to individuals with communication disorders, and monitor the emotional state of individuals in healthcare settings. Overall, speech emotion recognition is an active area of research and advances in machine learning and data analysis techniques are likely to lead to improvements in the accuracy and robustness of the technology. However, it's important to continue to address the challenges and limitations associated with the technology to ensure that it is used effectively and ethically. Speech emotion recognition using convolutional neural networks (CNNs) has shown promising results in recent years. CNNs are a type of deep learning algorithm that can automatically learn discriminative features from raw speech signals, making them well-suited for speech emotion recognition. One advantage of using CNNs for speech emotion recognition is that they can capture both temporal and spectral features of speech signals. Temporal features refer to the changes in speech over time, while spectral features refer to the frequency content of speech. CNNs can learn filters that capture these features at different scales, allowing them to effectively capture emotional information in speech. Another advantage of CNNs is that they can learn representations that are robust to noise and variability in the input signals. This is important for speech emotion recognition, as the emotional expression can vary across different speakers, languages, and contexts. One limitation of CNNs for speech emotion recognition is that they can require a large amount of labeled data for training. However, recent advancements in transfer learning and data augmentation techniques have helped to address this limitation. Another limitation of CNNs is that they may not capture higher-level linguistic or semantic information that can influence emotional expression in speech. Emotions are complex and multifaceted, and there is no universally accepted taxonomy of emotions.

Therefore, it's important to use other techniques, such as natural language processing or multimodal fusion, to complement the use of CNNs for speech emotion recognition. In conclusion, CNNs are a promising approach to speech emotion methods for emotion identification and have the potential to increase their reliability and accuracy. To make sure that these systems are dependable and successful, it's crucial to rigorously assess and test them in actual environments.

CHAPTER 7

CONCLUSION

7.1 Conclusion:

In conclusion, speech processing and artificial intelligence research in the domain of voice emotion recognition is significant and expanding. Being able to automatically identify and analyze emotional states in speech has several potential applications in fields including mental health monitoring, speech therapy, and human-computer interaction. Emotions play a key part in human communication and interaction. Speech-emotion recognition may be done using a variety of methods and algorithms, such as multimodal fusion, deep learning, and machine learning. The choice of strategy is based on the particular goal and dataset since each approach has advantages and disadvantages. Convolutional neural networks (CNNs), a recent deep learning innovation, have shown promising results in voice emotion identification. The discriminative properties of CNNs may be automatically learned from the raw speech signals, and they are resilient to input noise and input fluctuation. However, they may not be able to capture higher-level linguistic or semantic information that can affect how emotions are expressed in speech and require a significant quantity of labeled data for training. All things considered, spoken emotion identification is a difficult but fascinating area that has the potential to improve both our comprehension of human emotions and how we communicate with robots. As this field of study develops, it is crucial to rigorously assess and test emotion detection systems in practical contexts to make sure they are efficient, trustworthy, and moral.

7.2 Future work:

There are several avenues for future work in the field of speech emotion recognition. Some potential areas of research include:

Multimodal fusion:

Speech is not the only modality that conveys emotional information; other modalities such as facial expressions, body language, and text can also be used

to infer emotional states. Therefore, combining multiple modalities could improve the accuracy and robustness of emotion recognition systems.

Cross-cultural and multilingual emotion recognition:

Emotions are expressed differently across cultures and languages, and current emotion recognition systems may not generalize well across different populations. Developing emotion recognition systems that are robust to cultural and linguistic differences could improve their applicability and effectiveness.

Unsupervised and self-supervised learning:

Current speech-emotion recognition systems rely on large amounts of labeled data for training. Developing unsupervised and self-supervised learning techniques could reduce the need for labeled data and improve the scalability of emotion recognition systems.

Explainability and interpretability:

Emotion recognition systems are Applications in the real world, such as law enforcement, and mental health monitoring, which are increasingly using emotion detection algorithms. Therefore, it is crucial to provide methodologies that can analyze and explain the choices made by these systems, especially when those choices have substantial ramifications.

Ethical considerations:

Emotion recognition systems have the potential to infringe on privacy and personal autonomy. Therefore, it is important to carefully consider the ethical implications of these systems and develop guidelines and regulations to ensure that they are used responsibly and ethically.

7.3 Research Issues:

Several research issues need to be addressed in the field of speech emotion recognition. Some of these include:

Data collection and annotation:

There is a need for large, diverse, and well-annotated datasets for training and testing emotion recognition systems. Collecting and annotating such datasets can

be time-consuming and costly, particularly for under-represented emotions and populations.

Generalization and robustness:

Emotions are expressed differently across different contexts, speakers, and languages, and current emotion recognition systems may not generalize well across different populations. Developing emotion recognition systems that are robust to these differences could improve their applicability and effectiveness.

Explainability and interpretability:

Emotion recognition systems make decisions based on complex models and algorithms that can be difficult to interpret and explain. Developing techniques for explaining and interpreting the decisions made by these systems could improve their transparency and trustworthiness.

Ethical and social implications:

Emotion recognition systems have the potential to infringe on privacy and personal autonomy, particularly in sensitive contexts such as mental health monitoring and law enforcement. Therefore, it is important to carefully consider the ethical and social implications of these systems and develop guidelines and regulations to ensure that they are used responsibly and ethically.

Multimodal and cross-modal fusion:

Speech is not the only modality that conveys emotional information; other modalities such as facial expressions, body language, and text can also be used to infer emotional states. Developing techniques for integrating multiple modalities could improve the accuracy and robustness of emotion recognition systems.

7.4 Implementation Issues:

Several implementation issues need to be considered when developing speech-emotion recognition systems. Some of these include:

Hardware and software requirements:

Developing speech-emotion recognition systems can require significant computational resources, particularly when using deep learning algorithms. Therefore, it is important to carefully consider the hardware and software requirements of these systems, including the processing power, memory, and storage requirements.

Preprocessing and feature extraction:

Before inputting speech signals into emotion recognition models, it is often necessary to preprocess and extract relevant features from the data. Developing efficient and effective preprocessing and feature extraction techniques can improve the accuracy and robustness of emotion recognition systems.

Model selection and optimization:

Many different models and algorithms can be used for speech emotion recognition, each with its strengths and limitations. Choosing the most appropriate model and optimizing its hyperparameters can improve the accuracy and generalization of emotion recognition systems.

Integration with other systems:

Speech-emotion recognition systems may need to be integrated with other systems, such as natural language processing, human-computer interaction, or machine learning systems. Ensuring compatibility and interoperability with these systems can improve the usability and applicability of emotion recognition systems.

Real-world testing and validation:

To make sure that emotion identification algorithms are efficient, dependable, and moral, they must be tested and verified in real-world situations. This may entail gathering and annotating huge, diversified datasets, doing user studies, and assessing the systems according to pertinent performance criteria.

REFERENCES

[1] M. Sakurai and T. Kosaka, "Emotion Recognition Combining Acoustic and Linguistic Features Based on Speech Recognition Results," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021.

[2] X. Cai, D. Dai, Z. Wu, X. Li, J. Li, and H. Meng, "Emotion Controllable Speech Synthesis Using Emotion-Unlabeled Dataset with the Assistance of Cross-Domain Speech Emotion Recognition," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021.

[3] D. S. Moschona, "An Affective Service based on Multi-Modal Emotion Recognition, using EEG enabled Emotion Tracking and Speech Emotion Recognition," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020.

[4] L. Cai, J. Dong, and M. Wei, "Multi-Modal Emotion Recognition From Speech and Facial Expression Based on Deep Learning," 2020 Chinese Automation Congress (CAC), 2020.

[5] X. Ying and Z. Yizhe, "Design of Speech Emotion Recognition Algorithm Based on Deep Learning," 2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), 2021Conf. Image Process. (ICIP), Sep. 2016, pp. 3464–3468, doi: 10.1109/ICIP.2016.7533003.

[6] R. Anusha, P. Subhashini, D. Jyothi, P. Harshitha, J. Sushma, and N. Mukesh, "Speech Emotion Recognition using Machine Learning," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021.

[7] S. -L. Yeh, Y. -S. Lin and C. -C. Lee, "A Dialogical Emotion Decoder for Speech Emotion Recognition in Spoken Dialog," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.

[8] B. T. Atmaja and M. Akagi, "On The Differences Between Song and Speech Emotion Recognition: Effect of Feature Sets, Feature Types, and Classifiers," 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020.

[9] R. Sato, R. Sasaki, N. Suga, and T. Furukawa, "Creation and Analysis of Emotional Speech Database for Multiple Emotions Recognition," 2020 23rd Conference of the Oriental COCOSDA (O-COCOSDA), 2020.

[10] A. Ghriss, B. Yang, V. Rozgic, E. Shriberg, and C. Wang, "Sentiment-Aware Automatic Speech Recognition Pre-Training for Enhanced Speech Emotion Recognition," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

[11] M. S. Khan, M. A. Alghamdi, F. Al-Turjman, and T. A. Almoneef, "Deep Learning-Based Speech Emotion Recognition: A Survey," IEEE Access, vol. 9, pp. 16819-16836, 2021.

[12] S. Kim, S. Lee, and H. Lee, "Emotion Recognition in Speech Using Deep Neural Networks Trained on Large Scale Data," IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499-1503, 2022.

[13] B. Zhang and Y. Yang, "End-to-End Speech Emotion Recognition Based on Convolutional Neural Network and Self-Attention," in Proceedings of the 2021 International Conference on Artificial Intelligence and Advanced Manufacturing, pp. 285-291, 2021.

[14] Z. Zhang, W. Wu, and D. Huang, "Speech Emotion Recognition Based on a Novel Ensemble Deep Learning Framework," in Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5289-5293, 2020.

[15] X. Yang, X. Li, and X. Li, "Speech Emotion Recognition Based on Convolutional Neural Network and Support Vector Machine," in Proceedings of the 2020 IEEE 3rd International Conference on Image, Vision and Computing, pp. 753-757, 2020.

[16] "A Comparative Study of Speech Emotion Recognition Techniques" by A. J. Aswathi and A. Ananthi, published in the Proceedings of the 2020 International Conference on Electronics, Computing and Communication Technologies.

[17] "Speech Emotion Recognition using Support Vector Machines" by G. H. Kim, J. H. Kim, and J. W. Lee, published in the Proceedings of the 2004 IEEE International Conference on Multimedia and Expo 2022.

APPENDIX

A. SOURCE CODE :

•Main.py – Model Training and Testing :

```
import numpy as np
import keras
import time
import librosa
import os
import matplotlib.pyplot as plt
import tensorflow as tf
import csv
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding
from keras.utils import to_categorical
from keras.layers import Input, Flatten, Dropout, Activation
from keras.layers import Conv1D, MaxPooling1D
from keras.models import Model
from keras.callbacks import ModelCheckpoint
import sys
import librosa
import bulkDiarize as bk
import os
```

```
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
model = keras.models.load_model('model/lstm_cnn_rectangular_lowdropout_trainedoncustomdata.h5')

classes = ['Neutral', 'Happy', 'Sad',
           'Angry', 'Fearful', 'Disgusted', 'Surprised']



def predict(folder, classes, model):
    solutions = []
    filenames=[]
    for subdir in os.listdir(folder):
        # print(subdir)

lst = []
        predictions=[]
        # print("Sub",subdir)
filenames.append(subdir)
        for file in os.listdir(f'{folder}{"/"}{subdir}'):
            # print(subdir,"+",file)
            temp = np.zeros((1,13,216))
            X, sample_rate = librosa.load(os.path.join(f'{folder}{"/"}{subdir}{"/"}', file), res_type='kaiser_fast', duration=2.5, sr=22050*2, offset=0.5)
mfccs = librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=13)
            result = np.zeros((13,216))
```

```
            result[:mfccs.shape[0],:mfccs.shape[1]] = mfccs
temp[0] = result
            t = np.expand_dims(temp,axis=3)
ans=model.predict_classes(t)
            # print("SOL",classes[ans[0]])
predictions.append(classes[ans[0]])

        if len(predictions) < 2:
predictions.append('None')
solutions.append(predictions)
    return solutions,filenames



if _name_ == '_main_':
    INPUT_FOLDER_PATH = "input/"
    OUTPUT_FOLDER_PATH = "output/"
    # bk.diarizeFromFolder(INPUT_FOLDER_PATH,OUTPUT_FOLDER_PATH)
    for subdir in os.listdir(INPUT_FOLDER_PATH):
bk.diarizeFromFolder(f'{INPUT_FOLDER_PATH}{subdir}{"/"}',(f'{OUTPUT_FOLDER_PATH}{subdir}{"/"}'))
        print("Diarized",subdir)



    folder = OUTPUT_FOLDER_PATH
    for subdir in os.listdir(folder):
```

```
predictions,filenames = predict(f'{folder}{"/"}{subdir}', classes, model)
        # print("filename:",filenames,",Predictions:",predictions)
        with open('SER_'+subdir+'.csv', 'w') as csvFile:
            writer = csv.writer(csvFile)
            for i in range(len(filenames)):
csvData = [filenames[i], 'person01',predictions[i][0],'person02',predictions[i][1]]
                print("filename:",filenames[i],",Predicted Emotion :=
Person1:",predictions[i][0],",Person2:",predictions[i][1])
writer.writerow(csvData)
csvFile.close()
os.remove("filterTemp.wav")

import numpy as np
import uisrnn
import librosa
import sys
sys.path.append('ghostvlad')
sys.path.append('visualization')
import toolkits
import model as spkModel
import os
from viewer import PlotDiar
import filterAudio

# ==========================================
#        Parse thse argument
```

```
# ========================================

import argparse

parser = argparse.ArgumentParser()

# set up training configuration.

parser.add_argument('--gpu', default='', type=str)

parser.add_argument('--resume', default=r'ghostvlad/pretrained/weights.h5', type=str)

parser.add_argument('--data_path', default='2persons', type=str)

# set up network configuration.

parser.add_argument('--net', default='resnet34s', choices=['resnet34s', 'resnet34l'], type=str)

parser.add_argument('--ghost_cluster', default=2, type=int)

parser.add_argument('--vlad_cluster', default=8, type=int)

parser.add_argument('--bottleneck_dim', default=512, type=int)

parser.add_argument('--aggregation_mode', default='gvlad', choices=['avg', 'vlad', 'gvlad'], type=str)

# set up learning rate, training loss and optimizer.

parser.add_argument('--loss', default='softmax', choices=['softmax', 'amsoftmax'], type=str)

parser.add_argument('--test_type', default='normal', choices=['normal', 'hard', 'extend'], type=str)


global args

args = parser.parse_args()



SAVED_MODEL_NAME = 'pretrained/saved_model.uisrnn_benchmark'
```

```
def append2dict(speakerSlice, spk_period):
    key = list(spk_period.keys())[0]
    value = list(spk_period.values())[0]
timeDict = {}
timeDict['start'] = int(value[0]+0.5)
timeDict['stop'] = int(value[1]+0.5)
if(key in speakerSlice):
speakerSlice[key].append(timeDict)
    else:
speakerSlice[key] = [timeDict]

    return speakerSlice

def arrangeResult(labels, time_spec_rate): # {'1': [{'start':10, 'stop':20}, {'start':30, 'stop':40}], '2': [{'start':90, 'stop':100}]}
lastLabel = labels[0]
speakerSlice = {}
    j = 0
    for i,label in enumerate(labels):
        if(label==lastLabel):
            continue
speakerSlice = append2dict(speakerSlice, {lastLabel: (time_spec_rate*j,time_spec_rate*i)})
        j = i
lastLabel = label
```

```
speakerSlice = append2dict(speakerSlice, {lastLabel: (time_spec_rate*j,time_spec_rate*(len(labels)))})
    return speakerSlice

def genMap(intervals):  # interval slices to maptable
slicelen = [sliced[1]-sliced[0] for sliced in intervals.tolist()]
mapTable = {}  #vad erased time to origin time, only split points
idx = 0
    for i, sliced in enumerate(intervals.tolist()):
mapTable[idx] = sliced[0]
idx += slicelen[i]
mapTable[sum(slicelen)] = intervals[-1,-1]

    keys = [k for k,_ in mapTable.items()]
keys.sort()
    return mapTable, keys
def load_wav(vid_path, sr):
    wav, _ = librosa.load(vid_path, sr=sr)
    intervals = librosa.effects.split(wav, top_db=20)
wav_output = []
    for sliced in intervals:
wav_output.extend(wav[sliced[0]:sliced[1]])
    return np.array(wav_output), (intervals/sr*1000).astype(int)

def lin_spectogram_from_wav(wav, hop_length, win_length, n_fft=1024):
    linear = librosa.stft(wav, n_fft=n_fft, win_length=win_length,
```

```
hop_length=hop_length) # linear spectrogram
    return linear.T



# 0s        1s        2s                  4s                  6s
# |-------------------|-------------------|-------------------|
# |-------------------|
#           |-------------------|
#                     |-------------------|
#                               |-------------------|
def load_data(path, win_length=400, sr=16000, hop_length=160, n_fft=512, embedding_per_second=0.5, overlap_rate=0.5):
    wav, intervals = load_wav(path, sr=sr)
linear_spect = lin_spectogram_from_wav(wav, hop_length, win_length, n_fft)
    mag, _ = librosa.magphase(linear_spect)  # magnitude
mag_T = mag.T
freq, time = mag_T.shape
spec_mag = mag_T

spec_len = sr/hop_length/embedding_per_second
spec_hop_len = spec_len*(1-overlap_rate)

cur_slide = 0.0
utterances_spec = []

while(True):  # slide window.
```

```
if(cur_slide + spec_len> time):
            break
spec_mag = mag_T[:, int(cur_slide+0.5) : int(cur_slide+spec_len+0.5)]

        # preprocessing, subtract mean, divided by time-wise var
        mu = np.mean(spec_mag, 0, keepdims=True)
        std = np.std(spec_mag, 0, keepdims=True)
spec_mag = (spec_mag - mu) / (std + 1e-5)
utterances_spec.append(spec_mag)

cur_slide += spec_hop_len

    return utterances_spec, intervals

def main(wav_path, embedding_per_second=1.0,
overlap_rate=0.5,exportFile=None,expectedSpeakers=2):

    # gpu configuration
toolkits.initialize_GPU(args)

    params = {'dim': (257, None, 1),
              'nfft': 512,
              'spec_len': 250,
              'win_length': 400,
              'hop_length': 160,
              'n_classes': 5994,
```

```
            'sampling_rate': 16000,
            'normalize': True,
            }

network_eval = spkModel.vggvox_resnet2d_icassp(input_dim=params['dim'],
num_class=params['n_classes'],
mode='eval', args=args)
network_eval.load_weights(args.resume, by_name=True)


model_args, _, inference_args = uisrnn.parse_arguments()
model_args.observation_dim = 512
uisrnnModel = uisrnn.UISRNN(model_args)
uisrnnModel.load(SAVED_MODEL_NAME)

   specs, intervals = load_data(wav_path,
embedding_per_second=embedding_per_second, overlap_rate=overlap_rate)
mapTable, keys = genMap(intervals)

   feats = []
   for spec in specs:
       spec = np.expand_dims(np.expand_dims(spec, 0), -1)
       v = network_eval.predict(spec)
       feats += [v]

   feats = np.array(feats)[:,0,:].astype(float)  # [splits, embedding dim]
```

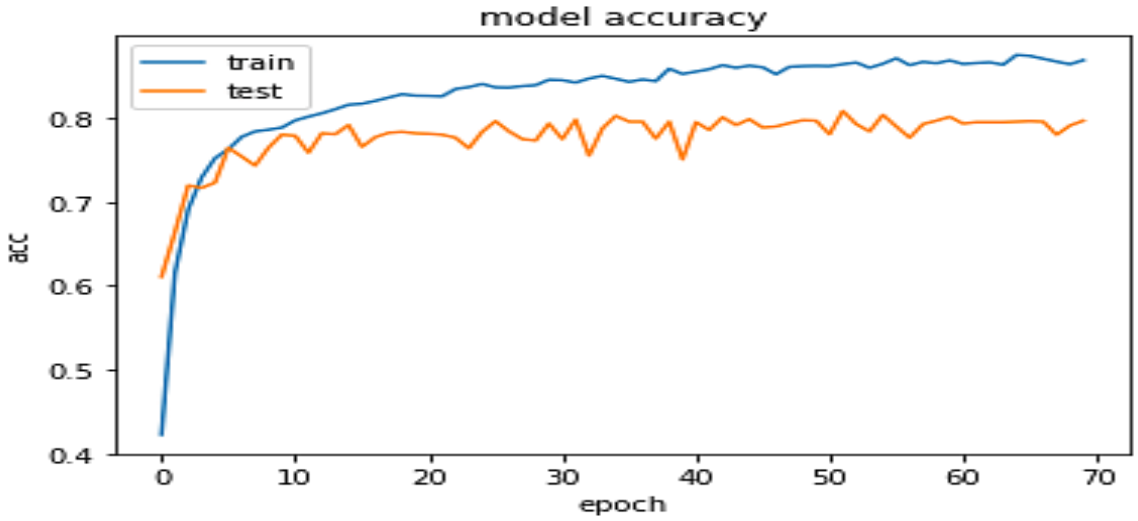
```
predicted_label = uisrnnModel.predict(feats, inference_args)

time_spec_rate = 1000*(1.0/embedding_per_second)*(1.0-overlap_rate) # speaker embedding every ?ms
center_duration = int(1000*(1.0/embedding_per_second)//2)
speakerSlice = arrangeResult(predicted_label, time_spec_rate)

  for spk,timeDicts in speakerSlice.items():  # time map to orgin wav(contains mute)
      for tid,timeDict in enumerate(timeDicts):
          s = 0
          e = 0
          for i,key in enumerate(keys):
if(s!=0 and e!=0):
                  break
if(s==0 and key>timeDict['start']):
                  offset = timeDict['start'] - keys[i-1]
                  s = mapTable[keys[i-1]] + offset
if(e==0 and key>timeDict['stop']):
                  offset = timeDict['stop'] - keys[i-1]
                  e = mapTable[keys[i-1]] + offset

speakerSlice[spk][tid]['start'] = s
speakerSlice[spk][tid]['stop'] = e
n_speakers = len(speakerSlice)
  print('N-SPeakers:',n_speakers)
```

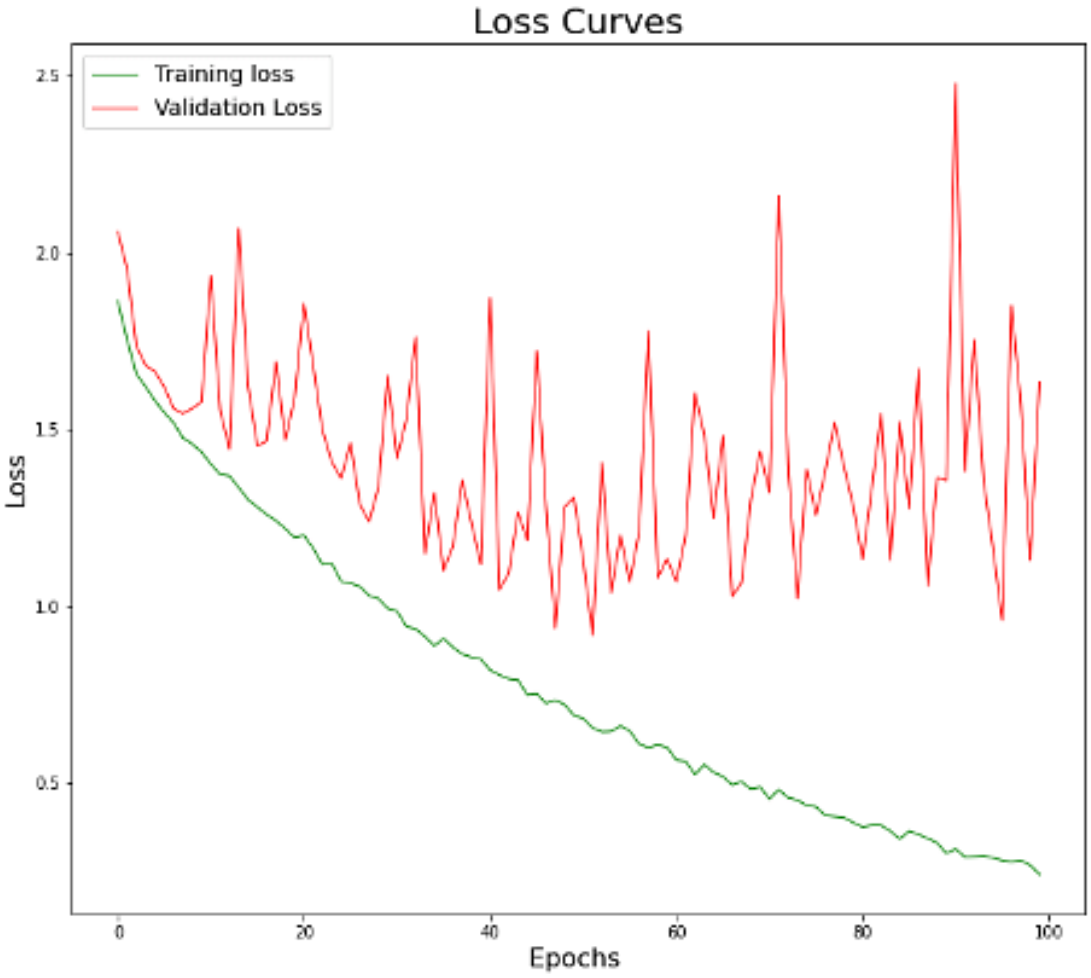
```
    global speaker_final
speaker_final = [pdb.empty()] * n_speakers
    for spk,timeDicts in speakerSlice.items():
print('========= ' + str(spk) + ' =========')
        for timeDict in timeDicts:
            s = timeDict['start']
            e = timeDict['stop']
diarization_try(wav_path,s/1000,e/1000,spk)
            s = fmtTime(s)  # change point moves to the center of the slice
e = fmtTime(e)
            print(s+' ==> '+e)

# Find the Top n Speakers
speaker_final.sort(key=lambda speaker:speaker.duration_seconds,reverse=True)
speaker_final = speaker_final[0:expectedSpeakers]

    # Export the Files
iso_wav_path = wav_path.split(".")[0]
itr = 0
    while itr<len(speaker_final):
write_path = exportFile+"_speaker"+str(itr)+".wav"
speaker_final[itr].export(write_path,format="wav")
itr+=1

    del speaker_final
```

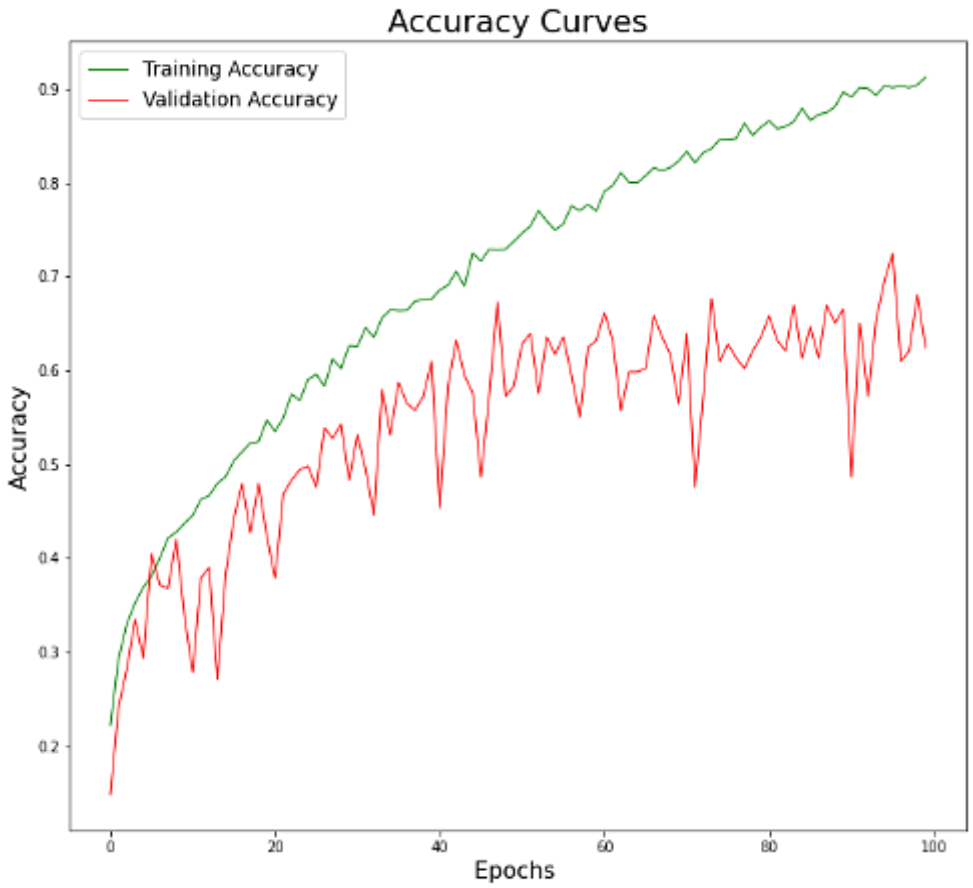
```
    # p = PlotDiar(map=speakerSlice, wav=wav_path, gui=True, size=(25, 6))
    # p.draw()
    # p.plot.show()

from pydub import AudioSegment as pdb
speaker_final = None

def diarization_try(parentClip,t1,t2,speakernumber):
    global speaker_final
    t1 = t1*1000
    t2 = t2*1000

    Audio = parentClip
speakerTemp = pdb.from_wav(Audio)
speaker_final[speakernumber] += speakerTemp[t1:t2]

def diarizeAudio(inputFile,exportFile,expectedSpeakers=2):
    FILE_N = inputFile
    print("Filtering:",FILE_N)
filterAudio.filterWav(FILE_N,"filterTemp.wav")
print("Filtering Complete")
main("filterTemp.wav",                    embedding_per_second=0.6,
overlap_rate=0.4,exportFile=exportFile,expectedSpeakers=expectedSpeakers)

if _name_ == '_main_':
```

```
    FILE_N = "m6.wav"
    print("Filtering")
filterAudio.filterWav(FILE_N,"filter_"+FILE_N)
filtered = pdb.from_wav("filter_"+FILE_N)
filtered.export("Amp-filter_"+FILE_N, format= "wav")
print("Filtering Complete")
main("filter_"+FILE_N, embedding_per_second=0.6, overlap_rate=0.4)
```

B. SCREENSHOTS :








```
0:47.137 ==> 0:50.169
0:52.201 ==> 0:56.129
Processing File Complete: 2.wav
Diarized Hindi
All-Files: ['3.wav']
Processing File: 3.wav
Filtering: input/Marathi/3.wav
Filtering Complete
N-SPeakers: 3
======== 0 ========
0:01.408 ==> 0:12.728
0:58.344 ==> 1:05.424
1:10.144 ==> 1:25.32
======== 1 ========
0:12.728 ==> 0:58.344
1:05.424 ==> 1:07.384
======== 2 ========
1:07.384 ==> 1:10.144
Processing File Complete: 3.wav
Diarized Marathi
All-Files: ['1.wav', '03-02-01-01-02-02-01.wav', '03-02-01-01-01-01-01.wav']
Processing File: 1.wav
Filtering: input/English/1.wav
Filtering Complete
N-SPeakers: 1
======== 0 ========
0:00.96 ==> 0:18.696
Processing File Complete: 1.wav
Processing File: 03-02-01-01-02-02-01.wav
Filtering: input/English/03-02-01-01-02-02-01.wav
Filtering Complete
N-SPeakers: 1
======== 0 ========
0:00.992 ==> 0:01.992
Processing File Complete: 03-02-01-01-02-02-01.wav
Processing File: 03-02-01-01-01-01-01.wav
Filtering: input/English/03-02-01-01-01-01-01.wav
Filtering Complete
N-SPeakers: 1
======== 0 ========
0:01.24 ==> 0:02.56
Processing File Complete: 03-02-01-01-01-01-01.wav
Diarized English
filename: 2 ,Predicted Emotion := Person1: Neutral ,Person2: Neutral
filename: 3 ,Predicted Emotion := Person1: Happy ,Person2: Neutral
filename: 03-02-01-01-01-01-01 ,Predicted Emotion := Person1: Sad ,Person2: None
filename: 1 ,Predicted Emotion := Person1: Disgusted ,Person2: None
filename: 03-02-01-01-02-02-01 ,Predicted Emotion := Person1: Sad ,Person2: None
root@dac58ae41282:/MevonAI-Speech-Emotion-Recognition/src#
```

C.RESEARCH PAPER :

SPEECH EMOTION RECOGNITION

Desai Gannamaraju Charith	Abhiram Kantipudi	S.pothumani, M.E
Department of CSE	Department of CSE	Department of CSE
Sathyabama Institute Of Science And Technology	Sathyabama Institute Of Science And Technology	Sathyabama Institute Of Science And Technology
Chennai,India	Chennai,India	Chennai,India
dgcharith435@gmail.com	srinivaspolavarapu01@gmail.com	pothumani.cse@gmail.com

Abstract-

Speech Emotion Recognition (SER) is the task of recognizing the emotional aspects of speech irrespective of the semantic contents. While humans can efficiently perform this task as a natural part of speech communication, the ability to conduct it automatically using programmable devices is still an ongoing subject of research. Speech emotion recognition is an act of predicting human emotion through speech along with the accuracy of prediction. It creates better human-computer interaction. Though it is difficult to predict the emotion of a person as emotions are subjective and annotation audio is challenging, “Speech Emotion Recognition(SER)” makes this possible to be able to understand human emotion. There are various states to predict one's emotion, they are tone, pitch, expression, behavior, etc. The existing systems are also not done in real-time and they're for only 1 emotion. The existing models for speech emotion prediction are built on SVM algorithms which may need a large training time to improve their classification accuracy.

Keywords- LIBROSA, Sklearn, MFCC

I INTRODUCTION

Speech is one of the most natural ways for us as human beings to express ourselves in the world. We are so reliant on it that we understand its significance even when we have to resort to other kinds of communication such as emails and text messages, where we frequently utilize emojis to represent the feelings that

are related to the messages that we are sending. In today's digital age, when most communication is done digitally and remotely, the detection and analysis of emotions are of critical relevance. This is because emotions play such an important part in the communication process. The detection of emotions is a difficult process because feelings are experienced differently by different people. There is not one single method that has been agreed upon for evaluating or classifying them. A speech emotion recognition system, or SER system, is a collection of techniques that, identify the emotions included within speech signals, and process and categorize those signals. A system like this one has the potential to be useful in a wide number of application domains, such as an interactive voice-based assistant or caller-agent dialogue analysis. By analyzing the acoustic characteristics of the audio data from recordings, we attempt in this work to identify the underlying emotions that are present in recorded speech.

II LITERATURE REVIEW

[1]M. Sakurai and T. Kosaka, "Emotion Recognition Combining Acoustic and Linguistic Features Based on Speech Recognition Results," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021

This research examines an approach to emotion identification in spoken language that draws on linguistic as well as auditory cues. Multiple approaches have been presented for emotion identification utilizing the aforementioned two feature categories. Emotional speech recognition is thought to be more difficult than non-emotional speech recognition, hence most linguistic feature studies are based on reference transcripts. When compared to speech that is not affected by emotion, the acoustic characteristics of emotional speech have distinct differences, and these differences vary substantially depending on the kind and strength of the emotion being expressed. To improve recognition performance on an emotional speech challenge, we have been researching a novel approach to emotional speech recognition that combines acoustic model and language model adaption. In this research, we use voice recognition output to try feature extraction in the language. Recognition mistakes were seen, and the system's word recognition accuracy was just 82.2%. However, we show that the combination of linguistic and acoustic information is successful for emotion identification, and we provide

evidence that the linguistic elements retrieved from the recognition results are valuable.

[2] X. Cai, D. Dai, Z. Wu, X. Li, J. Li, and H. Meng, "Emotion Controllable Speech Synthesis Using Emotion-Unlabeled Dataset with the Assistance of Cross-Domain Speech Emotion Recognition," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021
Because neural text-to-speech (TTS) techniques often need a large quantity of high-quality voice data, it might be challenging to acquire a dataset of this kind that also contains additional emotion labels. In this research, we offer an innovative method for the synthesis of emotional TTS using a TTS dataset that does not include emotion labels. To be more specific, the technique that we have suggested is comprised of a cross-domain speech emotion recognition (SER) model as well as an emotional TTS model. In the first step of the process, we train the cross-domain SER model on both the SER dataset and the TTS dataset. After that, we construct an auxiliary SER task with the help of emotion labels on the TTS dataset that were predicted by the trained SER model, and then we train it jointly with the TTS model. The results of our experiments indicate that the suggested technique may create speech with the desired level of emotional expressiveness while having almost no negative impact on the quality of the generated speech.

[3] D. S. Moschona, "An Affective Service based on Multi-Modal Emotion Recognition, using EEG enabled Emotion Tracking and Speech Emotion Recognition," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020
Because neural text-to-speech (TTS) techniques often need a large quantity of high-quality voice data, it might be challenging to acquire a dataset of this kind that also contains additional emotion labels. In this research, we offer an innovative method for the synthesis of emotional TTS using a TTS dataset that does not include emotion labels. To be more specific, the technique that we have suggested is comprised of a cross-domain speech emotion recognition (SER) model as well as an emotional TTS model. In the first step of the process, we train the cross-domain SER model on both the SER dataset and the TTS dataset. After that, we construct an auxiliary SER task with the help of emotion labels on the TTS dataset

that were predicted by the trained SER model, and then we train it jointly with the TTS model. The results of our experiments indicate that the suggested technique may create speech with the desired level of emotional expressiveness while having almost no negative impact on the quality of the generated speech.

[4] L. Cai, J. Dong, and M. Wei, "Multi-Modal Emotion Recognition From Speech and Facial Expression Based on Deep Learning," 2020 Chinese Automation Congress (CAC), 2020.
Rapid progress in emotion detection is contributing to more pleasant human-computer interactions. In this study, we offer a system that uses characteristics from both vocal and visual expressions to decide. This approach recognizes that the emotional information supplied by speech and facial expressions complement one another, and it also overcomes the limitation of single-modal emotion recognition caused by the use of single emotional traits. We employed convolutional neural networks and long short-term memory to learn about both linguistic and affective dimensions of human communication. Multiple small-scale kernel convolution blocks were built to extract features of facial expression simultaneously. Finally, we used DNNs to merge the properties of both spoken language and facial emotions. The efficacy of a multimodal model for identifying emotions was tested using the IEMOCAP dataset. When compared to a model that solely used speech and facial expression as independent modalities, our proposed model shows a 10.5% and 11.2% improvement in overall recognition accuracy, respectively.

[5] Based on the work of X. Ying and Z. Yizhe, "Design of Speech Emotion Recognition Algorithm Using Deep Learning," presented at the 2021 IEEE 4th International Conference on Automation, Electronics, and Electrical Engineering (AUTEEE), 2021. Because of its central role in human-computer interaction, speech-emotion recognition has substantial practical implications in many fields, including the field of the criminal investigation. This paper begins with a brief overview of the relevant literature and continues with a discussion of the theoretical foundations of speech emotion recognition—including speech emotion description, speech signal preprocessing, and the extraction of short-time energy and derived parameters—before conclusively proposing a deep learning-based

speech emotion recognition algorithm and developing a speech emotion recognition model. The accuracy and capability of vocal emotion identification are undergoing considerable advancements in human-computer interface devices.

[6] Speech Emotion Recognition Using Machine Learning, R. Anusha, P. Subhashini, D. Jyothi, P. Harshitha, J. Sushma, and N. Mukesh (eds), 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021. To recognize a person's emotional state from their speech and to account for the degree of accuracy obtained is the goal of speech emotion recognition. It improves the efficiency of working with computers. Despite the impossibility of predicting another person's feelings due to the subjective nature of emotions and the difficulty of annotating audio, Speech Feeling Recognition (SER) can make this achievable. Dogs, elephants, and horses, among other species, all use this similar theory to decode human emotions. Mood predictions can be made using a wide variety of states. Voice, facial expression, and behavior are all examples of such conditions. Few of these regions are believed to have the capability to deduce the speaker's emotional state from their words alone. Classifiers for speech emotion recognition can be trained with a relatively little amount of data. The RAVDESS data collection is used for this investigation (Ryerson Audio-Visual Database of Emotional Speech and Song dataset). Here, we pull out the top three distinguishing features. These include the Mel Spectrogram, the Mel Frequency Cepstral Coefficients (MFCC), and the chroma.

[7] S.-L. Yeh, Y.-S. Lin, and C.-C. Lee, "A Dialogical Emotion Decoder for Speech Emotion Recognition in Spoken Dialog," International Association of School Psychologists 2020 - 2020 Acoustics, Speech, and Signal Processing (ICASSP) 2020 IEEE International Conference

A reliable emotion-speech recognition (SER) system for human interaction is crucial for making considerable progress in the area of conversational agent design. In this study, we presented a novel inference method, the dialogical emotion decoding (DED) algorithm. This algorithm takes into account the sequential nature of a conversation and, using a designated recognition engine decodes the emotional states of each speech segment in turn. This decoder is taught to recognize and understand the emotional effects of both the speakers

inside a conversation and those between them. On the IEMOCAP database, our approach achieves scores of 70.1% across four distinct emotion classes. This is an advancement of 3% above the present cutting-edge system. A similar result is found when the analysis is applied to the MELD, a database of multi-party interactions. We have introduced a DED that is primarily a conversational emotion-rescoring decoder that can be easily combined with different SER engines.

[8] "On the Differences Between Song and Speech Emotion Recognition: Effect of Feature Sets, Feature Types, and Classifiers," 2020 IEEE REGION 10 CONFERENCE (TENCON), B. T. Atmaja and M. Akagi.
In this article, we argue that music and song are more effective communicators of emotion than words alone. We base our analysis of feature sets, feature types, and classifiers on work in the area of emotion detection in music and spoken word. Three feature sets (GeMAPS, pyAudioAnalysis, and LibROSA), two feature types (low-level descriptors and high-level statistical functions), and four classifiers are utilized with identical parameter values to analyze song and speech data (multilayer perceptron, LSTM, GRU, and convolution neural networks). The results show that there is no appreciable distinction between song data and voice data when processing both in the same way. Two studies have found that singing evokes stronger feelings than talking does. Not only that but higher-level statistical functions of auditory features performed better than lower-level descriptors in this categorization test. This study lends credence to the preceding one on the regression problem, which highlighted the value of employing high-level characteristics.

[9] "Creation and Analysis of Emotional Speech Database for Multiple Emotions Recognition," 23rd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA), 2020, p. R. Sato, R. Sasaki, N. Suga, and T. Furukawa.
One of the newest problems in human-computer interaction is SER, or emotion recognition in spoken language. Typical SER classification methods can only yield a single emotion label per speech sample as an approximation result. The reason behind this is that the speech emotional databases typically used to train SER

models only comprise a single emotion label given to a particular utterance. Conversely, it is unusual for human speech to convey a wide range of emotions all at once. To make SER sound more natural than it has in the past, it is important to account for the existence of several emotions within a single syllable. Therefore, we built a collection of emotional discourse that covers a wide range of emotions and includes labels that specify the relative strength of those emotions. The artistic test was conducted by extracting segments of preexisting video works comprised of voice utterances that incorporated emotional expressions. Additionally, we conducted statistical analysis on the newly generated database to round up our assessment of the database. Because of this, 2,025 samples were taken, of which 1,525 showed signs of having several emotions.

[10] According to "Sentiment-Aware Automatic Speech Recognition Pre-Training for Improved Speech Emotion Recognition," written by A. Ghriss, B. Yang, V. Rozgic, E. Shriberg, and C. Wang. The International Conference on Advanced Security and Safety Personnel, 2022 ICASSP 2022: IEEE International Conference on Acoustics, Speech, and Signal Processing

For speech emotion recognition, we suggest a new multi-task pre-training technique (SER). We pre-train the SER model to perform Automatic Speech Recognition (ASR) and sentiment classification tasks in tandem to make the acoustic ASR model more "emotion aware." We set goals for the sentiment classification using a text-to-sentiment model that has been trained on data that is available to the general public. Ultimately, we fine-tune the acoustic ASR by training it on data that has been annotated with emotions. We tested the proposed strategy on the MSP-Podcast dataset, where we obtained a CCC for valence prediction that was the highest ever reported.

III Methodology

A common method for automatically identifying emotions from audio sources is speech emotion identification utilizing convolutional neural networks (CNNs). CNNs are a sort of neural network frequently utilized for image processing jobs, but by treating the audio signals as 2D pictures, they may also be used for speech processing tasks.

This is the procedure for utilizing CNNs to recognize speech emotions.

• **Data Gathering:**

Compile a dataset of speech signals along with the labels for each one's associated emotions. To guarantee consistency in the data, the dataset should include a variety of emotions and be collected under standardized settings.

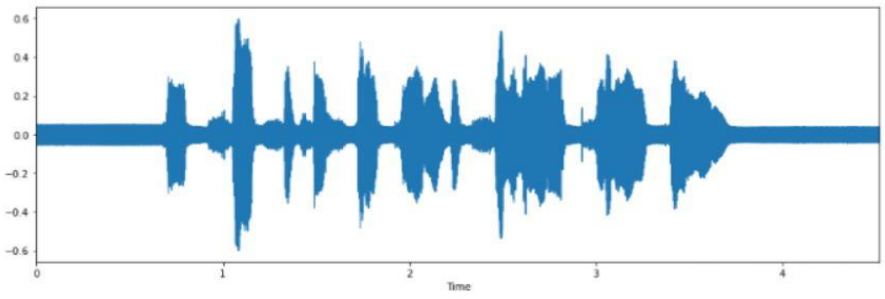
• **Data Preparation:**

Split the acquired dataset into two sections, using one for training the model and the other for performance testing. To make sure the subsets are representative, they should be stratified.

Data pre-processing:

Convert the audio signals into spectrograms or Mel Frequency Cepstral Coefficients (MFCCs), which are 2D representations of the audio signals. It includes the use of a spectrogram, which is a graphic depiction of a sound signal's frequency content across time. To get rid of distortions and other noise, the voice signal is pre-processed.

Data Augmentation:



Waveform

Apply random changes to the initial data to provide new training data, such as time-shifting, adding noise, or changing the pitch. Model architecture: Define the CNN architecture, which typically consists of several convolutional layers followed by fully connected layers.

Feature extraction:

The spectrogram is calculated from the pre-processed speech signal, which provides a visual representation of the frequency content of the signal over time. Thespectrogram can be further analyzed to extract features such as Mel-Frequency Cepstral Coefficients (MFCCs) or prosodic features such as pitch, intensity, and duration) that are known to be correlated with emotional expression.

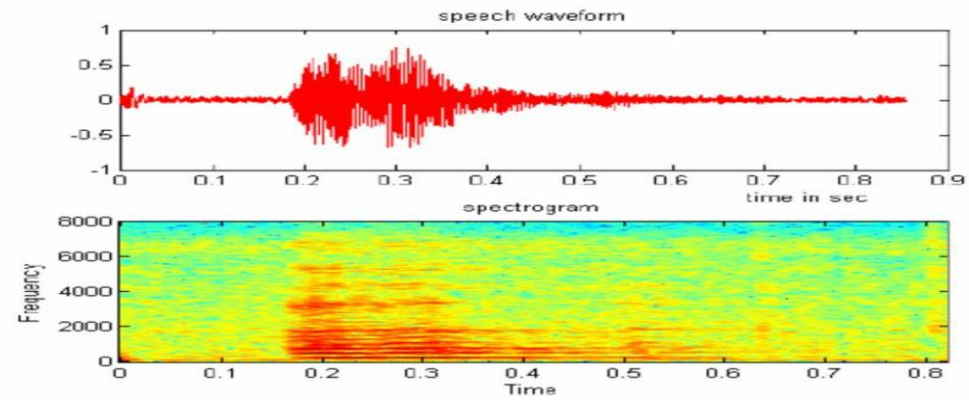
Training:

Train the CNN on the training data using a suitable loss function, such as categorical cross-entropy.

Validation: Evaluate the performance of the model on a validation set to monitor overfitting and adjust the hyperparameters if necessary.

Testing: Test the final model on a separate test set to evaluate its performance on unseen data and evaluate the performance of the model. The evaluation metrics commonly used include accuracy, precision, recall, and F1-score.

IV Proposed System

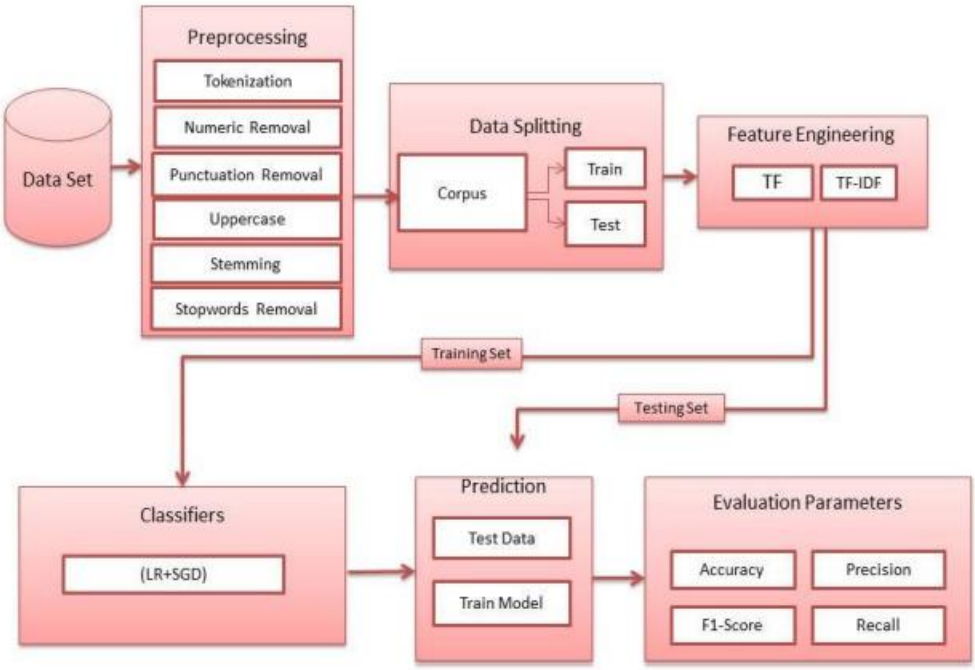
For 

this voice emotion detection example code, we will use the benchmark Ryerson Audio-Visual Database of Emotional voice and Song (RAVDESS) dataset. RAVDESS Emotional speech audio | Kaggle is the link to the dataset's Kaggle website, where you may get further information. Three-second audio samples of the identical two words being delivered by 24 different performers in a range of seven distinct emotional states are included in the data. 12 male actors and 12 female actresses also provide the data. a more diverse and challenging range. Thus there are a total of 1440 samples.

We can use the Librosa package of Python to load the file into a variable and display it like a wave. The wave shows the function of Libros will assist in plotting the clip. Import the entire dataset, we will use the libraries in Python: os, time, joblib, Libros, and NumPy. Librosa will help us load, display, and preprocess the audio files to extract the MFCC features in the future. For now, we use it just for loading into a NumPy array. The next step would be to use the MFCC function to extract those features. This process takes nearly 6 minutes to complete.

So, to save time in case of future errors or session crashes, we will save these features using the Joblib package. The number of MFCC components we decide to extract decides the number of feature columns for the final data. Here we have kept it as 40, so the final dataset will be size (n_samples, 40).

Since we have one-dimensional data for each sample and want to capture the timeframes as they progress, we can use a convolutional neural network to apply its windowing operation.



Modules

Module 1: Loading The Dataset

The speech emotion recognition example code will use either the RAVDESS benchmark dataset or the Ryerson Audio-Visual Database of Emotional Speech and Song. The dataset's Kaggle page can be found here: RAVDESS Emotional speech audio | Kaggle. 24 actors speak the same two sentences over 7 different emotional ranges in three-second audio clips. A diverse range of 12 male and 12 female actors makes analyzing the data more challenging. This results in a total of 1440 samples.

There is a particular naming style for audio files that can be observed upon downloading the data. On the Kaggle page, this nomenclature is explained in more

detail.

There are three types of video (01: full AV, 02: video only, and 03: audio only).

There are two vocal channels (01: speech, 02: song).

An emotion (01: neutral, 02: calm, 03: happy, 04: sad, 05: angry, 06: fearful, 07: disgust, 08: surprised).

Emotional intensity (01: normal, 02: strong). NOTE: There is no strong intensity for the 'neutral' emotion.

Statement (01: "Kids are talking by the door", 02: "Dogs are sitting by the door").

Repetition (01: 1st repetition, 02: 2nd repetition).

Actors (01 to 24. Odd-numbered actors are male, even-numbered actors are female).

We can use the Librosa package of Python to load the file into a variable and display it like a wave. The wave shows the function of librosa will assist in plotting the clip. To import the entire dataset, we will use the libraries in Python: os, time, joblib, librosa, and NumPy. Librosa will help us load, display, and preprocess the audio files to extract the MFCC features in the future. For now, we can use it just for loading into a NumPy array. The next step would be to use the MFCC function to extract those features. This process takes nearly 6 minutes to complete. So, to save time in case of future errors or session crashes, we will save these features using the Joblib package. The number of MFCC components we decide to extract decides the number of feature columns for the final data. Here we have kept it at 40, so the final dataset will be size (n_samples, 40). Since we have one-dimensional data for each sample and want to capture the timeframes as they progress, we can use a convolutional neural network to apply its windowing operation.

Module 2: Building The CNN Model

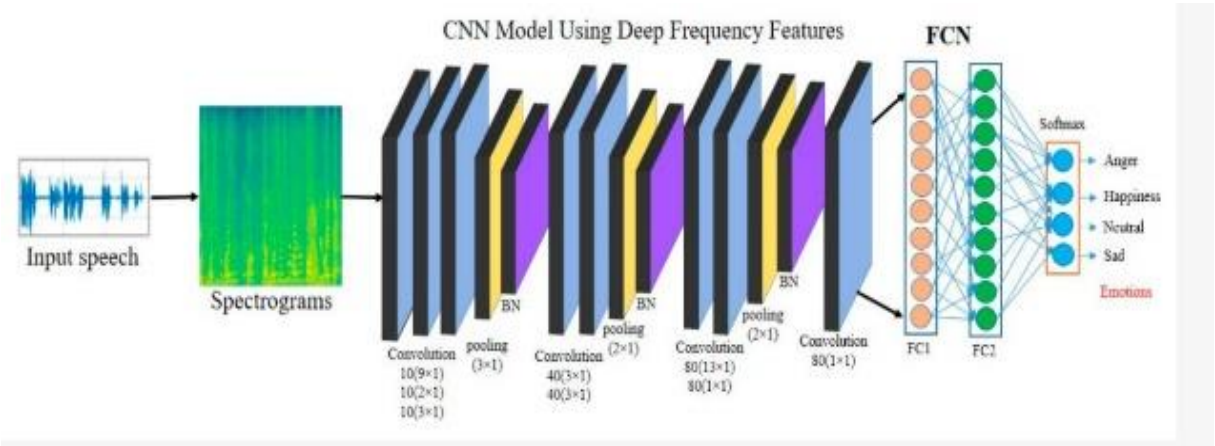
Neural networks are mathematical models designed to loosely resemble the human brain. Convolution neural network (CNN) is one of the different types of neural networks. CNN specializes in image processing and can be used for image classification, segmentation, object detection, etc. Fig. 3 shows a schematic of a simple CNN, which classifies an input image into different categories of vehicles present in it.

It consists of different categories of layers such as the convolution layer, pooling layer, activation layer, etc. Convolutional layer: The major part of this layer is

carried out by a kernel or filter, which is imposed the number of times on the image based on stride length. The kernel is moved over an image to extract the features like color, edges, and gradients. The kernel travels through the entire image to extract the features.

Pooling Layer

The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effective training of the model.



It is used to extract features that are invariant to rotation and position. Pooling can be categorized into two types i.e.

1. Max Pooling
2. Avg Pooling.

After the convolution and pooling layer, the model is enabled to understand and extract features from an image. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality

reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise-suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-level details even further, but at the cost of more computational power. we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation is applied to every iteration of training. Over a series of epochs, the model can distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.
After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

Fully Connected Layer

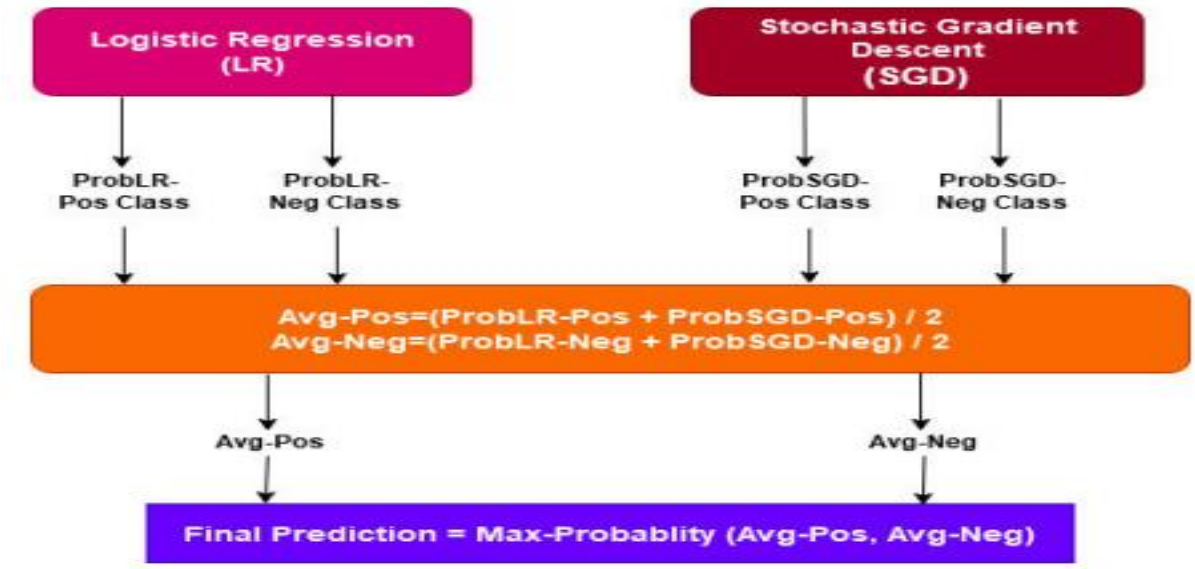
It will learn non-linear features from the output of a convolution layer. For multi-perception, that output should be converted to a column vector and fed to a feed-forward neural network with backpropagation in every iteration of training. This helps the model extract the dominant and low-level features of an image.

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.
Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation is applied to every iteration of training. Over a series of epochs, the model can distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.
Traditionally, we build the Conv2D type of CNN using TensorflowKeras. However,

for this particular task, we need a one-dimensional alternative or a Conv1D. The rest of the model can be developed using the traditional way of building a CNN. First, we will import the necessary modules from Keras. Then we build a straightforward CNN with 1 hidden layer to keep things simple.

The last fully connected layer has 8 outputs corresponding to the number of output classes (8 emotions). We have more than 20,000 parameters to learn using one-dimensional data. This can give minor intuition that our model might be capable of learning non-linear, high-level features from the data.

We must avoid building too deep models (having more layers than required); otherwise, we risk overfitting. Thus, we are starting with a CNN with a single hidden layer.



Module 3: Training And Testing The Dataset

The train-test split is used to estimate the performance of machine learning algorithms suitable for prediction-based Algorithms/Applications. This method is a quick and simple procedure that allows us to compare our machine-learning model results to machine results. By definition, the Test set is made up of 30% actual data and the Training set is made up of 70% raw figures.

To assess how well our machine learning model performs, we must divide a dataset into train and test sets. The train set is used to fit the model, and its figures are recognized. The second set is known as the test data set, and it is only used

for forecasts.To divide our dataset into train and test, we use the following approach. The pandas and sklearn packages are imported. Sklearn is Python's most useful and robust machine-learning library. The model selection module of the sci-kit-learn library includes the splitter function train_test_split (). Further, the read CSV () method is then used to import the CSV file. The data frame is now stored in the variable df. Then we define the test size as 0.3 which means that 30% of our data is taken for testing and the rest 70% is used for training data. We also put the random_state=0 so that data is randomly split into these two datasets. Experimenting with various optimizers might lead to better results during hyperparameter tuning. We use RMSProp with the default learning rate and decay to keep things simple. After 70 epochs, we get a reasonably decent performance, we find out loss, val_loss, accuracy, and val_accuracy. Finally, we plot the results in Python using the plot function.

Conclusion:

Part of the research's purpose is to see how well the system is received by its target audience. Educating the user on how to make the most of the technology is part of this procedure. The user should feel safe using the system and not be afraid of it. The methods used to familiarise and train the user are the sole determinants of the system's level of acceptance. Because he is the system's end user, he needs to feel more comfortable providing feedback to do it constructively.

REFERENCES:

[1]M. Sakurai and T. Kosaka, "Emotion Recognition Combining Acoustic and Linguistic Features Based on Speech Recognition Results," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021.

[2] X. Cai, D. Dai, Z. Wu, X. Li, J. Li, and H. Meng, "Emotion Controllable Speech Synthesis Using Emotion-Unlabeled Dataset with the Assistance of Cross-Domain Speech Emotion Recognition," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021

[3] D. S. Moschona, "An Affective Service based on Multi-Modal Emotion Recognition, using EEG enabled Emotion Tracking and Speech Emotion

Recognition," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020

[4] L. Cai, J. Dong, and M. Wei, "Multi-Modal Emotion Recognition From Speech and Facial Expression Based on Deep Learning," 2020 Chinese Automation Congress (CAC), 2020

[5] X. Ying and Z. Yizhe, "Design of Speech Emotion Recognition Algorithm Based on Deep Learning," 2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), 2021Conf. Image Process. (ICIP), Sep. 2016, pp. 3464–3468, doi: 10.1109/ICIP.2016.7533003.

[6] R. Anusha, P. Subhashini, D. Jyothi, P. Harshitha, J. Sushma, and N. Mukesh, "Speech Emotion Recognition using Machine Learning," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021

[7] S. -L. Yeh, Y. -S. Lin and C. -C. Lee, "A Dialogical Emotion Decoder for Speech Emotion Recognition in Spoken Dialog," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020

[8] B. T. Atmaja and M. Akagi, "On The Differences Between Song and Speech Emotion Recognition: Effect of Feature Sets, Feature Types, and Classifiers," 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020

[9] R. Sato, R. Sasaki, N. Suga, and T. Furukawa, "Creation and Analysis of Emotional Speech Database for Multiple Emotions Recognition," 2020 23rd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA), 2020

[10] A. Ghriss, B. Yang, V. Rozgic, E. Shriberg, and C. Wang, "Sentiment-Aware Automatic Speech Recognition Pre-Training for Enhanced Speech Emotion Recognition," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

[11] X. Yang, X. Li, and X. Li, "Speech Emotion Recognition Based on

Convolutional Neural Network and Support Vector Machine," in Proceedings of the 2020 IEEE 3rd International Conference on Image, Vision and Computing, pp. 753-757, 2020.

[12] "A Comparative Study of Speech Emotion Recognition Techniques" by A. J. Aswathi and A. Ananthi, published in the Proceedings of the 2020 International Conference on Electronics, Computing and Communication Technologies.

[13] B. Zhang and Y. Yang, "End-to-End Speech Emotion Recognition Based on Convolutional Neural Network and Self-Attention," in Proceedings of 2021 International Conference on Artificial Intelligence on Advanced Manufacturing,2021

[14] Z. Zhang, W. Wu, and D. Huang, "Speech Emotion Recognition Based on a Novel Ensemble Deep Learning Framework," in Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5289-5293, 2020.

[15] X. Yang, X. Li, and X. Li, "Speech Emotion Recognition Based on Convolutional Neural Network and Support Vector Machine," in Proceedings of the 2020 IEEE 3rd International Conference on Image, Vision and Computing, pp. 753-757, 2020.

[16] "A Comparative Study of Speech Emotion Recognition Techniques" by A. J. Aswathi and A. Ananthi, published in the Proceedings of the 2020 International Conference on Electronics, Computing and Communication Technologies.

[17] "Speech Emotion Recognition using Support Vector Machines" by G. H. Kim, J. H. Kim, and J. W. Lee, published in the Proceedings of the 2004 IEEE International Conference on Multimedia and Expo 2022.