USER AUTHENTICATED PERSONALIZED VOICE ASSISTANT

A PROJECT

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

Bachelor of Technology

by

Abijit Singh , Pranav Singh Sambyal
(2019BITE007, 2019BITE016)

Under the guidance

of

Dr. Iqra Altaf Gillani

DEPARTMENT OF INFORMATION TECHNOLOGY
NATIONAL INSTITUTE OF TECHNOLOGY
SRINAGAR - 190006 (INDIA)

June 2023

Certificate

We hereby certify that the work which is being presented in the project titled **User Authenticated Personalized Voice Assistant** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology and submitted in the Department of Information Technology, National Institute of Technology Srinagar, is an authentic record of our own work carried out during a period from July 2022 to June 2023 under the supervision of Dr. Iqra Altaf Gillani, Assistant Professor, Department of Information Technology, National Institute of Technology Srinagar.

The matter presented in this project report has not been submitted by me for the award of any other degree of this or any other Institute/University.

Abijit Singh **Pranav Singh Sambyal**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Iqra Altaf Gillani
Department of Information Technology

The project Viva-Voce Examination of **Abijit Singh** and **Pranav Singh Sambyal**, has been held on .....

Signature of Supervisor Signature of External Examiner

Acknowledgement

It is our privilege to express our sincerest regards to our department head **Dr Shabir Ahmad Sofi** for encouraging and allowing us to present the project on the topic **User authenticated personalized voice assistant** at our department premises for the partial fulfillment of the requirements leading to the award of B.Tech degree.

We deeply express our sincere thanks to our project guide, **Dr. Iqra Altaf Gillani**, for her valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of this project.

We take this opportunity to express gratitude to all of the department faculty members for their help and support.

We also place on record, our sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture.

Abijit Singh (2019BITE007)

Pranav Singh Sambyal (2019BITE016)

Abstract

This brief overview presents a personalized voice assistant system that includes speaker verification for user identification and uses speech-to-text conversion for further classification. The main purpose of the system is to improve the user experience by providing personalized services while ensuring secure access. User authentication is achieved through speaker verification technology that accurately identifies and authenticates users based on their unique voice characteristics. After authentication, the system uses speech-to-text technology to convert the user's voice input into text form for efficient processing and analysis. It then uses natural language processing techniques to categorize and understand user commands and queries. By integrating speaker verification and speech-to-text classification techniques, the system can provide customized responses and personalized service. This overview highlights the importance of user authentication and voice processing in personalized voice assistants, highlighting their potential to improve user privacy, security, and overall user experience. The proposed system paves the way for future research on voice-based user authentication and personalized voice assistant technology.

Contents

List of Tables

List of Figures

Abbreviations

CNN	Convolutional Neural Network
DNN	Deep Neural Network
MFCC	Mel Frequency Cepstral Coefficients
IoT	Internet of Things
STT	Speech To Text
RDS	Relational Database Services (offered by AWS)

Chapter 1

Introduction

As voice assistants have become more widespread, we have seen an increase in our interaction with technology and everyday tasks. We have become accustomed to using intelligent virtual assistants like Siri, Google Assistant, and Alexa every day, offering voice-based assistance and seamless communication. It is, however, critical to ensure that these voice assistants are personalized and authenticated by their users. Speaker verification and speech-to-text technology can be integrated for a promising solution.

Based on the speaker's voice characteristics, biometric technology is used for speaker verification to authenticate their identity. Speaker verification systems distinguish authorized users from unauthorized users by analyzing factors such as pitch, rhythm, and timbre. Security applications are their primary focus, but their potential is much broader.

By embedding speaker verification into personalised voice assistants, we may construct user-authenticated systems that adapt to individual preferences, offering personalised replies and services. Once a speaker's identification has been validated, the assistant can use sophisticated speech-to-text algorithms to translate spoken words into written text. This allows for additional analysis and categorization, allowing for seamless interaction and personalised experiences.

The use of speaker verification and speech-to-text technologies in personalised voice assistants has various benefits. It improves security by ensuring that only authorised persons can access sensitive information or conduct certain tasks. Furthermore, it offers personalised experiences by adjusting the assistant's replies and suggestions depending on the recognised user's interests and history.

We investigate the underlying technologies of speaker verification and speech-to-text conversion in this academic study to investigate the notion of user-authenticated personalised voice assistants. We look at the challenges and potential of integrating these technologies, as well as existing methodologies and strategies, and evaluate their implications for improving the functionality and user experience of voice assistants. We hope to contribute to the progress of personalised voice assistants and create creative applications in a variety of areas, including smart homes, healthcare, virtual assistants in automobiles, and customer service, by grasping the potential of this technological fusion.

1.1 Problem Statement & Objectives

Our problem statement was to design a system, which can satisfy the following requirements:

1. *I*n today's fast-paced and technologically-driven world, most people rely on voice assistants for various tasks and information gathering. In spite of the fact that existing voice assistants are capable of providing a personalized experience, their effectiveness is hindered and limited..

2. It is necessary to develop and implement personalized voice assistants that adapt to users' individual needs by identifying their voices. These assistants will provide tailored responses, suggestions, and functionalities based on user preferences, habits, and contextual information to address this problem. To constantly refine and enhance their performance, these personalized voice assistants should be capable of learning and evolving over time

The objectives at the same time, were to ensure the following:

1. Speaker verification :Using a speaker verification technique, an individual can be identified or denied as being that person based on the characteristics of their unique voice. A voice sample provided by the individual is compared to a reference voice print or voice model that has been pre-recorded and can be used as a guide to determine the voice type. It is a representation of the acoustic features of a person's voice, which serve as their unique acoustic signature

2. Speech to text:Hugging Face includes a popular library known as ”transformers,” which contains pre-trained models for a variety of NLP applications, including voice recognition. These models are based on cutting-edge designs like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer), which have made substantial advances in language comprehension and creation. Hugging Face's pre-trained STT models, such as ”wav2vec” or ”deepspeech,” are trained on large-scale datasets to convert spoken language into written text and may be used to execute speech-to-text tasks. These models are intended to handle a wide range of languages and have been trained on large quantities of audio data to understand the patterns and characteristics of spoken language.

3. Understand and respond to text: By implementing this technology locally, it will be capable of acting as an intelligible voice assistant when only simple commands are given, as well as understanding and responding to texts given to it.

1.2 Organization

The rest of the report is organized as follows. We review the existing literature in Chapter 2. In chapter 3, we introduce the concepts behind Speaker verification. In chapter 4, we have mentioned the how we did feature extraction, the model used, and other details. In chapter 5, we discuss the implementation details of the project, explaining all its features, and specified the code snippets wherever required. In chapter 6, we have reviewed the current scenario of public about the significance of voice assistants in their lives . Finally in chapter 7, we have briefly summarized the project, and given directions for future work.

Chapter 2

Literature Survey

In this chapter, we survey the recent literature to gain some insights into the prevailing research which has been done in the field. Based on each paper, certain observations and inferences were made and thus our groundwork for the project was started.

Probing Deep Speaker Embeddings for Speaker-related Tasks by Zhao, Z., Pan, D., Peng, J., Gu R. (2022) [1] The notion of speaker embeddings, which are low-dimensional representations of speaker identities learnt by deep neural networks, is introduced in the study. The authors significantly increase speaker verification performance over conventional approaches by training a deep neural network to translate input voice signals to speaker embeddings. The research also investigates the applicability of speaker embeddings to tasks like speaker diarization and voice recognition. Overall, the study demonstrates the use of deep learning approaches in speaker verification and paves the way for future advances in speaker-related applications.

End-to-End Text-Independent Speaker Verification with Triplet Loss on Short Utterances by hang, C. Koishida, K. (2017 : Architecture, Consensus, Future Trends [2] The study focuses on the difficulties presented by text-dependent speaker verification, in which the speaker must pronounce certain words to be verified. To train discriminative speaker embeddings, the authors create a deep neural network architecture that integrates the triplet loss. When compared to standard approaches, the suggested method delivers considerable performance gains, especially in circumstances with little training data. The study also investigates the effect of various training

tactics and data augmentation approaches on verification performance. Overall, the study proposes an end-to-end solution for text-dependent speaker verification utilising the triplet loss that is both successful and efficient.

The notion of speaker embeddings, which are low-dimensional representations of speaker identities learnt by deep neural networks, is introduced in the study. The authors significantly increase speaker verification performance over conventional approaches by training a deep neural network to translate input voice signals to speaker embeddings. The research also investigates the applicability of speaker embeddings to tasks like speaker diarization and voice recognition. Overall, the study demonstrates the use of deep learning approaches in speaker verification and paves the way for future advances in speaker-related applications.

Improved Speaker Verification with Deep Residual Networks by Tang et al. (2018) : Architecture, Consensus, Future Trends [3] Deep residual networks are used in a unique way to improve speaker verification performance. The article overcomes the limits of standard speaker verification approaches by utilising the power of residual networks, which enable efficient deep representation learning. The authors present a residual network architecture developed expressly for speaker verification tasks, which they train on large-scale datasets to capture discriminative speaker embeddings. The experimental findings show that the suggested methodology surpasses existing methods in terms of accuracy and resilience in a variety of demanding settings. Furthermore, the research investigates the effect of various network designs and training methodologies on verification performance. Overall, the study adds to the area of speaker verification by employing deep residual networks to increase accuracy and dependability.

Attentive Speaker Verification: Learning Attention Mechanism for Speaker Verification Using Deep Neural Networks : Architecture, Consensus, Future Trends [4] incorporates an attention mechanism into deep neural networks to provide a fresh way to speaker verification. The research tackles the problem of dealing with variable-length utterances by employing an attention mechanism that dynamically focuses on key acoustic aspects. The suggested model may effectively capture speaker-specific features and improve verification performance by paying attention to relevant parts. Experiment findings on common benchmark datasets show that the attention

mechanism improves speaker verification accuracy, especially in circumstances with background noise and reverberation. The research also studies the influence of several versions of the attention mechanism on system performance. Overall, this study advances the area of speaker verification by presenting an attention-based strategy that enhances the robustness and accuracy of speaker verification systems.

Table 2.1: Literature Review

Name	Description	Year	Group
Probing Deep Speaker Embeddings for Speaker-related Tasks [1]	resents a neural network architecture that learns discriminative speaker embeddings from raw speech waveforms, enabling accurate verification.	2022	Zhao, Z., Pan, D., Peng, J. , Gu, R. .
End-to-End Text-Independent Speaker Verification with Triplet Loss on Short Utterances [2]	To learn discriminative speaker embeddings, the method employs both speaker and text information.	2017	Zhang, C., Koishida, K.
Dr-Vectors: Decision Residual Networks and an Im- proved Loss for Speaker Recognition [3]	Their approach utilizes residual learning to improve the performance of speaker verification systems, resulting in more accurate speaker embeddings.	2022	Pelecanos, J., Wang, Q. , Moreno, I.
Uttention Mechanism in Speaker Recognition: What Does It Learn in Deep Speaker Embedding? [4]	Deep neural networks with attention mechanisms are used in this technique to improve speaker embeddings.	2018	Wang, Q., Okabe, K. Lee, K., Yamamoto, H. , Koshinaka, T.

Chapter 3

Proposed system

This system aims to enhance accuracy and efficiency in voice-based interactions. User Speaker Verification ensures secure access by authenticating the speaker's identity. Speech-to-Text technology converts spoken language into written text. Text Classification categorizes transcribed text, aiding in information retrieval and decision-making processes.

Speaker verification, also known as voice recognition , is a technology used to verify a speaker's claimed identity based on their unique voice characteristics. This is a bio metric modality that identifies a person based on their voice characteristics. The speaker verification process typically includes her two main phases: Registration and Verification. During enrollment, a user's voice sample is captured and used to create a unique voice print or speaker model. This voice print represents the unique characteristics of an individual's voice, including aspects such as pitch, rhythm, rhythm, and pronunciation patterns. Voice prints are stored in a database for future reference. When the user provides the individuals ID during the verification phase, their voice is captured and compared to the stored voice print. The matching system analyzes the speech samples and calculates the similarity between the input speech and the enrolled voice prints. If the similarity value exceeds a predefined threshold, the user is considered authenticated and the claimed identity is confirmed. Otherwise, validation will be rejected. Speaker verification systems use a variety of techniques and algorithms to extract and analyze speech features from speech samples. A common approach is to use machine learning techniques such as Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), and Deep Neural Networks (DNN). These models learn to recognize and distinguish between different speakers by capturing characteristic patterns in speech data.

Several factors are considered to improve the performance of speaker verification systems. An important aspect is the quality of the voice data recorded during enrollment and verification. Factors such as background noise, channel variations, and voice variations can affect the accuracy of the verification process. To mitigate these challenges, we use preprocessing techniques and signal enhancement algorithms to clean and normalize the speech data.

Another important aspect is the need to prevent spoofing and fraudulent attempts to trick the system. An attacker could attempt unauthorized access by impersonating a legitimate speaker. To address this issue, advanced techniques such as liveliness detection, where the system verifies the presence of a live speaker, can be integrated into the speaker verification process. Speaker verification is used in many areas. It is commonly used in secure access control systems where voice is used as a bio metric identifier, rather than traditional methods such as passwords or ID cards. It is also used in telephone banking systems, voice assistants, and forensic investigations to authenticate individuals based on their voice characteristics.

Although speaker authentication technology has advanced, challenges still remain. Speech variations due to factors such as age, medical conditions, and emotional states can affect system accuracy. In addition, efficient storage and retrieval mechanisms are required to handle large deployments and manage large numbers of registered voice prints.

In summary, speaker verification is a bio metric technology that uses voice characteristics to verify people's claimed identities. Extract and analyze audio features using techniques such as machine learning, preprocessing, and signal amplification. With ongoing research and advancements, speaker verification is constantly evolving, providing a secure and convenient solution for identity verification in a variety of applications.

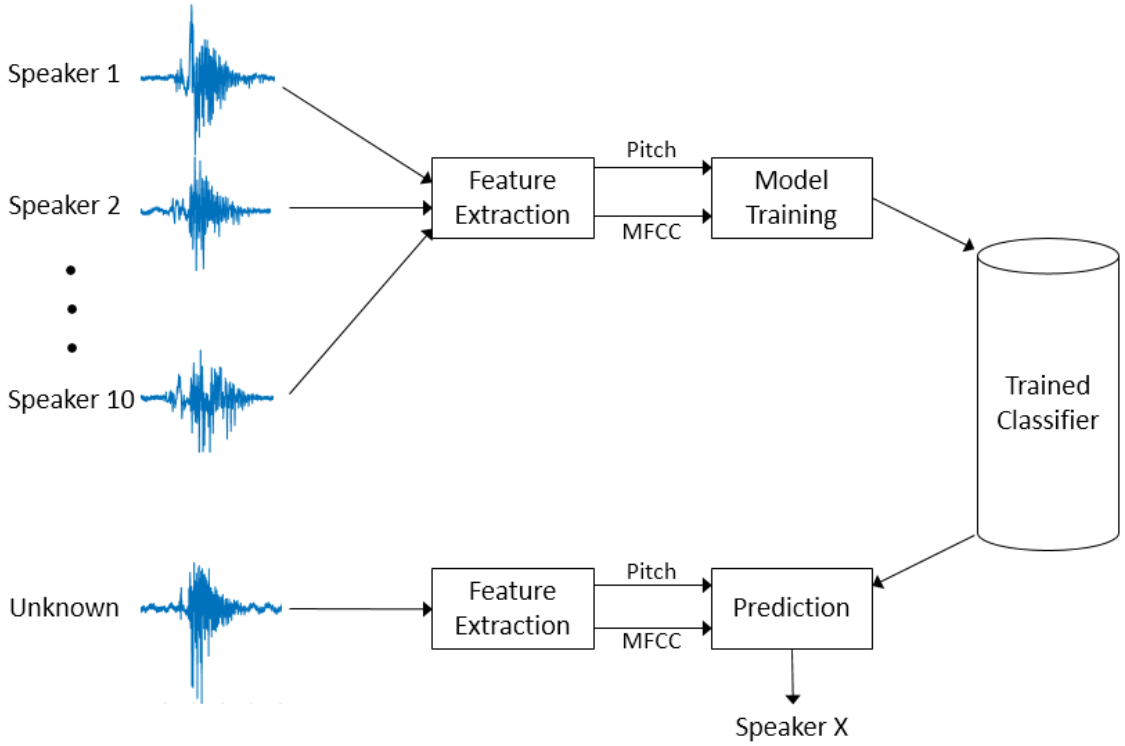

Figure 3.1: Visual representation of speaker verification

Following were the steps implemented in speaker verification:

1. Preprocessing audio files:

    - Audio files form the data set are organized and arranged in a fashion which makes it easy to perform further operations on it. Data's balance is also checked in this round to ensure that the data is not skewed in order to insure a well rounded data.
    - After importing the audio files, do the appropriate preprocessing actions. MFCC (Mel Frequency Cepstral Coefficients) and spectrograms may be extracted from audio files using LibROSA's utilities for importing audio files. Important facets of the speaker's voice are captured by these traits.

2. Feature Extraction:

    - Take relevant features from the preprocessed audio and extract them. MFCC is a characteristic that is often employed for speaker verification jobs. LibROSA may be used to compute MFCC coefficients from audio data.

- Tonal centroid features, also known as Tonnetz features, are a type of acoustic feature derived from the pitch information of speech signals and provide insights into the tonal characteristics of a speaker's voice.In speaker verification, the goal is to determine whether a given speaker's voice matches a reference voice or not. Tonal centroid features capture the distribution of pitch classes in a speech signal and can be used to represent the tonal structure of the voice

3. Model Training:

   - In most cases, speech verification entails training a machine learning model on a labelled data-set of voice samples.The features extracted in the previous step is used to train this model so it can later be used to make accurate pridictions.

4. Model Assessment:

   - Following the training of the model, use a distinct dataset to evaluate how well it performs after being trained. It is desirable for the dataset to include both audio samples from speakers that are recognised and imposter samples from speakers that are unknown. To evaluate the model's speaker verification success, compute metrics such as accuracy, precision, recall, and score.

5. Verification:

   - To perform speaker verification on a fresh audio sample, use the same procedure as in the training phase to extract the features. After that, run these characteristics through the trained model to get a prediction or similarity score. Determine if the sample corresponds to the claimed speaker or not using a predefined threshold.

6. Speech to text:

   - openai/whisper-base is a pre-trained model provided by OpenAI that is specifically designed for Speech-to-Text (STT) tasks. It is based on the Whisper ASR system, which stands for "Whisper Automatic Speech Recognition" Whisper is trained on a large amount of multilingual and multitask supervised data collected from the web.

- These models are trained on vast amounts of audio data so they can learn the patterns and features of spoken language. Using the Hugging Face API or SDK, developers can easily send audio files or streams to the model and receive the transcription as text output.

- Hugging Face's speech-to-text service is part of a broader focus on natural language processing (NLP) and aims to bridge the gap between speech-based and text-based applications. Hugging Face allows developers to create applications that provide automatic transcription, voice commands, voice assistants, and more.

- The Hugging Face library, called 'Transformers', offers a wide range of pre-trained models, including models for speech-to-text tasks. These models are based on state-of-the-art architectures such as BERT and GPT known for their advances in language understanding and generation.

- Speech-to-Text capabilities can be important in many areas, including transcription services, speech-enabled systems, customer support automation, and accessibility for the deaf. Hugging Face allows developers to rely on robust pre-trained models to accelerate the development of these applications. Hugging Face's speech-to-text capabilities evolve as NLP research advances and are regularly updated and improved. Developers can benefit from community contributions and collaborate with others to improve the functionality of their speech-to-text applications. Hugging Face also provides extensive documentation, tutorials, and samples to help developers effectively integrate speech-to-text functionality. This support helps you understand model parameters, fine-tune models for specific tasks, and optimize performance based on your specific requirements.

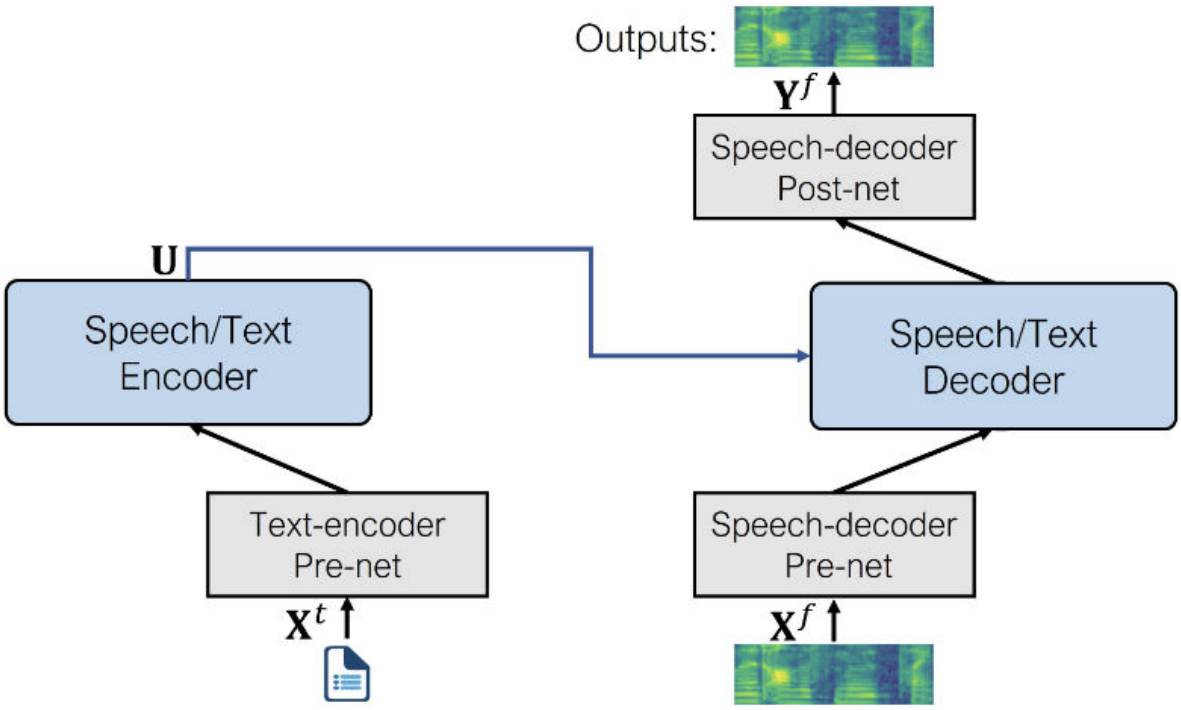

Figure 3.2: Visual representation of speech to text

Chapter 4

Implementation

This chapter elucidates the project overview, including the key technique and technologies used in the development of this project.

Feature Extraction:

```
def extractfeatures(files):
      # Loads files for folder into the function
    file  = os.path.join(os.path.abspath('Folder Containing data')+'/'+str(
       files.file))

    # Loads the audio file as a time series, with default rate
    X, rate = librosa.load(file, res_type='kaiser_fast')

    # Generates Mel-frequency cepstral coefficients (MFCCs) from a
        the abovetime series
    mfccs = np.mean(librosa.feature.mfcc(y=X, sr=rate, n_mfcc=40).T,axis=0)

    # Generates a Short-time Fourier transform (STFT) to be used
       in the chroma_stft
    stft = np.abs(librosa.stft(X))

```

```
    # Computes a chromagram from a power spectrogram.
    chroma = np.mean(librosa.feature.chroma_stft(S=stft, sr=rate).T,axis=0)

    # Computes the spectral contrast
    contrast = np.mean(librosa.feature.spectral_contrast(S=stft, sr=rate).T
        ,axis=0)

    # Computes a mel-scaled spectrogram.
    mel = np.mean(librosa.feature.melspectrogram(y=X, sr=rate).T,axis=0)


    # Computes the tonal centroid features used for geting tonal
        characters of the voics data (tonnetz)
    tonnetz = np.mean(librosa.feature.tonnetz(y=librosa.effects.harmonic(X)
        ,sr=rate).T,axis=0)

    # We also add the classes of each file as a label at the end
        1:Male 0:Female
    label = files.label

    return mfccs, chroma, mel, contrast, tonnetz, label
```

The extractfeatures function uses the capability of libraries like librosa to do various computations on audio input throughout the speaker identification process. It produces critical properties such as MFCCs, which capture spectral information, and chroma, which represents pitch distribution. Furthermore, the mel spectrogram, spectral contrast, and tonnetz characteristics provide additional information on the frequency content, spectrum differences, and tonal structure of the audio, respectively. These collected features are fed into machine learning models, which enable accurate speaker identification by learning unique patterns unique to each individual's voice.

Model used:

Sequential models in deep learning are linear stacks of layers used to build various neural network architectures. It is a fundamental concept and one of the simplest and most commonly used models in neural network design. A model consists of multiple layers that are added in sequence, with the output of one layer serving as the input to the next layer. This sequential nature allows a direct flow of data from input to output. Sequential models are commonly used to build feed-forward neural networks in which information flows in one direction. They are build and trained using popular deep learning frameworks such as Keras and TensorFlow. The order in which one add layers determines the architecture and information flow in your model. Sequential models are well suited for tasks such as image classification, text classification, and sequence prediction. They are especially effective when the input data has a well-defined structure. During training, model parameters are learned using gradient descent and backpropagation. A model is compiled with an optimizer, a loss function, and an evaluation metric. The optimizer determines the algorithm for updating the model weights, the loss function measures the difference between the predicted output and the actual output, and the evaluation metric provides an additional measure of performance. Models are trained by iteratively adjusting the weights to minimize the loss function using techniques such as stochastic gradient descent and mini-batch processing. The training process spans multiple epochs, with each epoch representing a pass through the training data set. Validation data allows you to monitor model performance using unseen data and perform early stopping if necessary. Overall, sequential models serve as the basic building blocks of deep learning, enabling the construction and training of powerful neural network architectures.

```
SeqModel = Sequential()

SeqModel.add(Dense(X.shape[1], X.shape[1], activation = 'relu'))
SeqModel.add(Dropout(0.1))

SeqModel.add(Dense(128, activation = 'relu'))
SeqModel.add(Dropout(0.25))

SeqModel.add(Dense(128, activation = 'relu'))
SeqModel.add(Dropout(0.5))
```

```
11
12 SeqModel.add(Dense(y.shape[1], activation = 'softmax'))
13
14 SeqModel.compile(loss='categorical_crossentropy', metrics=['accuracy'],
      optimizer='adam')
15
16 early_stop = EarlyStopping(monitor='val_loss', min_delta=0, patience=100,
      verbose=1, mode='auto')
```

The model is a five-layer Sequential model for speaker identification. It consists of an input layer, three hidden layers, each with 128 neurons and ReLU activation. To prevent overfitting, dropout layers are included, with dropout rates of 0.1, 0.25, and 0.5 in consecutive layers.The final layer is the output layer, which uses softmax activation for multi-class classification and has the same number of neurons as the number of classes in the speaker identification task. Because of this architecture, the model can learn complicated patterns and features in audio data, allowing for reliable speaker recognition.

The model is compiled using categorical_crossentropy as the loss function. The optimizer used is Adam, which is an efficient optimization algorithm. An EarlyStopping callback is defined to monitor the validation loss during training. It stops the training process if the validation lost does not improve for a certain number of epochs.

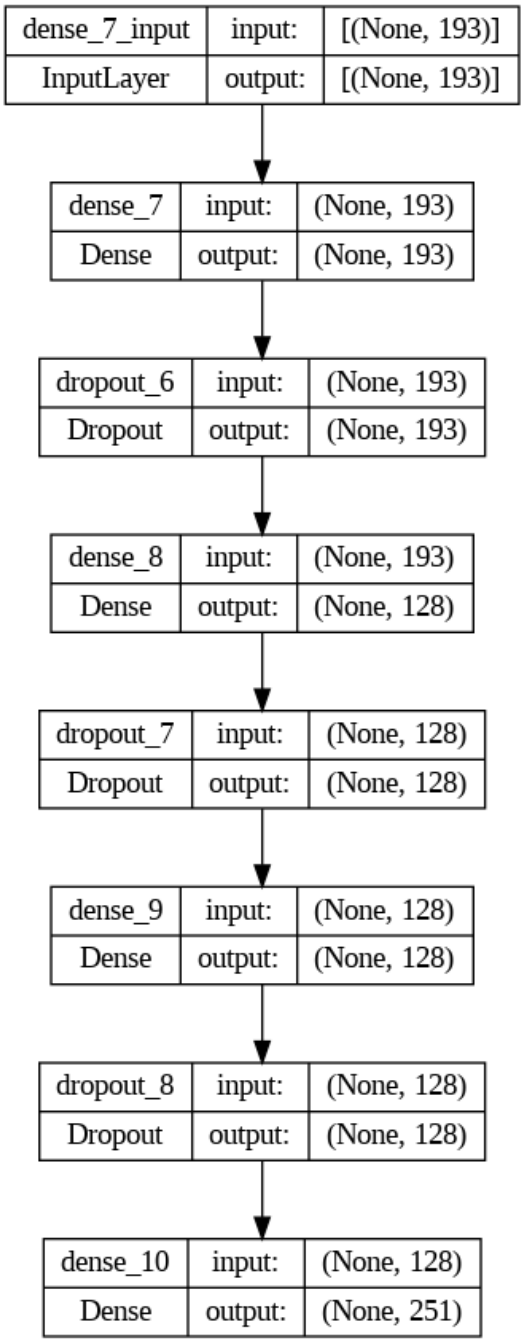

Figure 4.1: Model architecture diagram

Chapter 5

Results

Training and Validation accuracy by epoch:

We noticed a considerable improvement in accuracy and a steady decrease in loss over time in our investigation. With each iteration, the model's accuracy grew steadily, showing a stronger knowledge and prediction capability. Simultaneously, the loss reduced progressively, suggesting the model's capacity to match the data more successfully and minimise errors. The gains in accuracy and loss indicate that the model is learning and adapting well to the given dataset. The growing accuracy implies that the model's predictions are more closely aligned with the actual outcomes. Concurrently, lowering loss indicates that the model is reducing the difference between anticipated and actual values, leading in a more precise representation of the data. These findings show that the training method is beneficial, since the model constantly improves its prediction performance by increasing accuracy and decreasing loss. This advancement is due to the optimisation approaches and iterative learning algorithms used during the training phase.

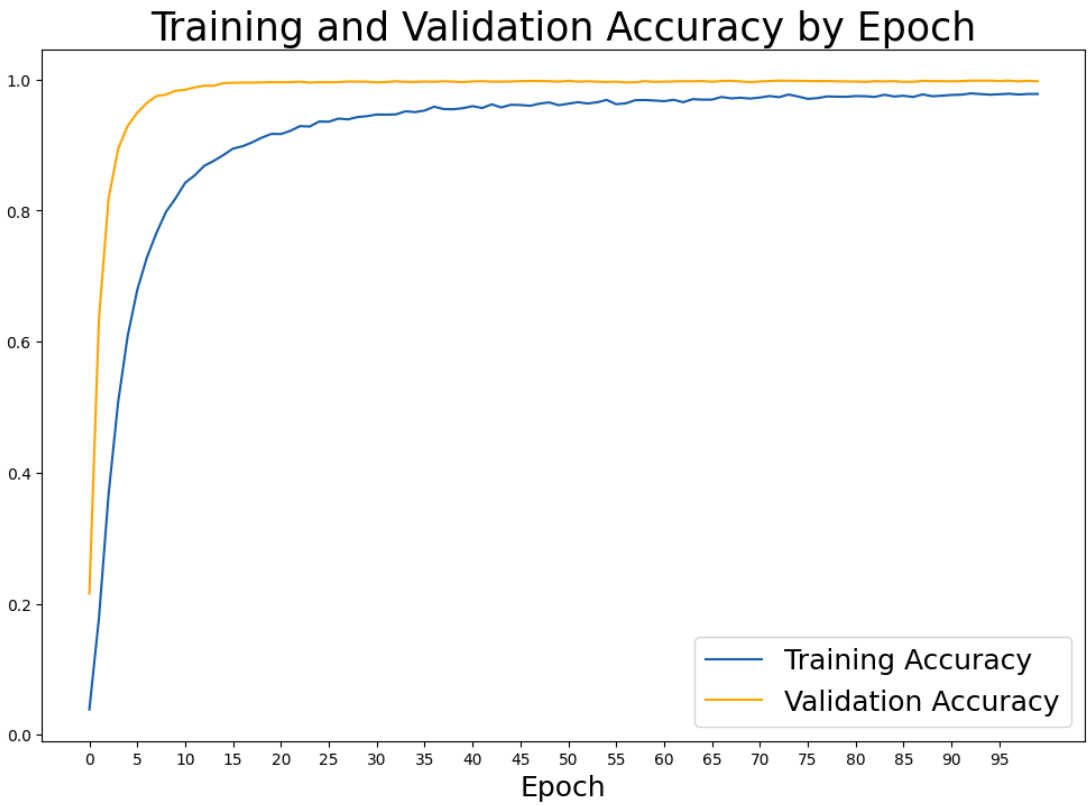

Figure 5.1: Graph of training and accuracy

Using the trained model to make an voice assistant app [9]:

This report introduces an innovative application that combines speaker verification, speech-to-text conversion, and intelligent action processing.The app's goal is to improve the user experience by offering secure login, accurate speech recognition, and dynamic action generation based on recognised text. The speaker verification methodology assures secure access, while speech-to-text conversion allows users to engage with the app by speaking. The retrieved text is subsequently processed in order to initiate particular activities, such as retrieving real-time news updates and delivering weather.Implementation and evaluation confirm the integrated approach's success in providing accurate authentication and dynamic action processing.

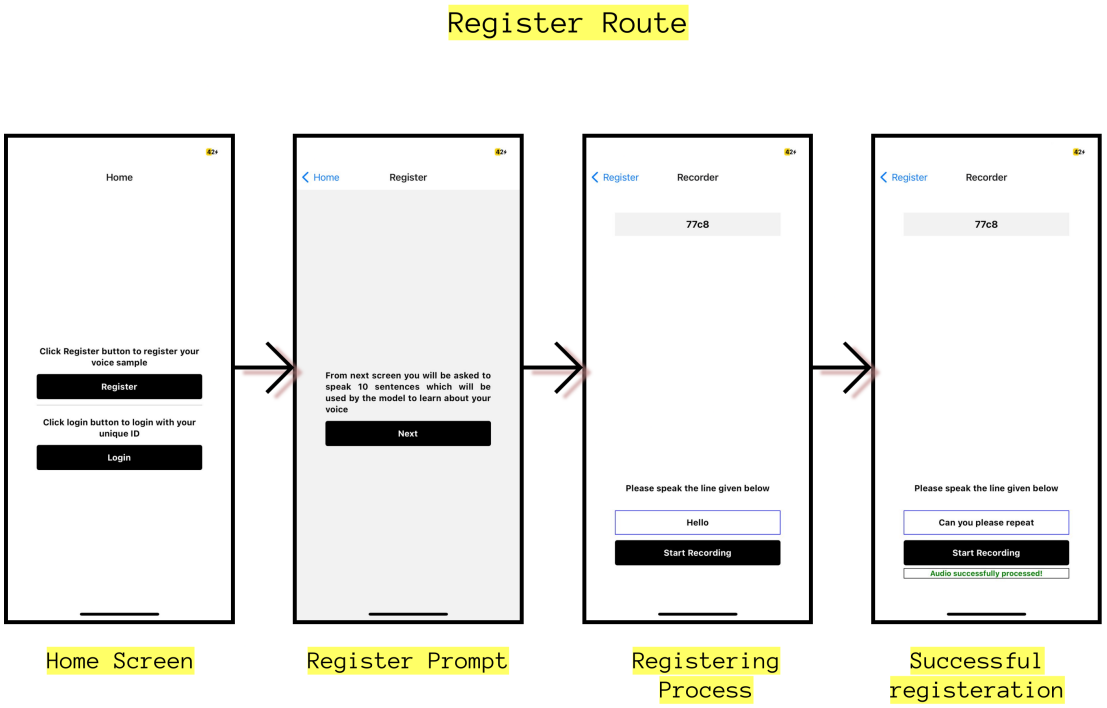

Figure 5.2: Register Route of app

The above diagram consists of the register route of the app. In this section the app tells the user that ,they need to say some lines of text so the model can understand their voice and recognize

them when given the task. After each successful recording the back-end responses with a 'Audio Processing Successful' text. Once a set number of samples are retrieved then the model is initiated to learn to understand the users voice and stored the result in a database for future reference.

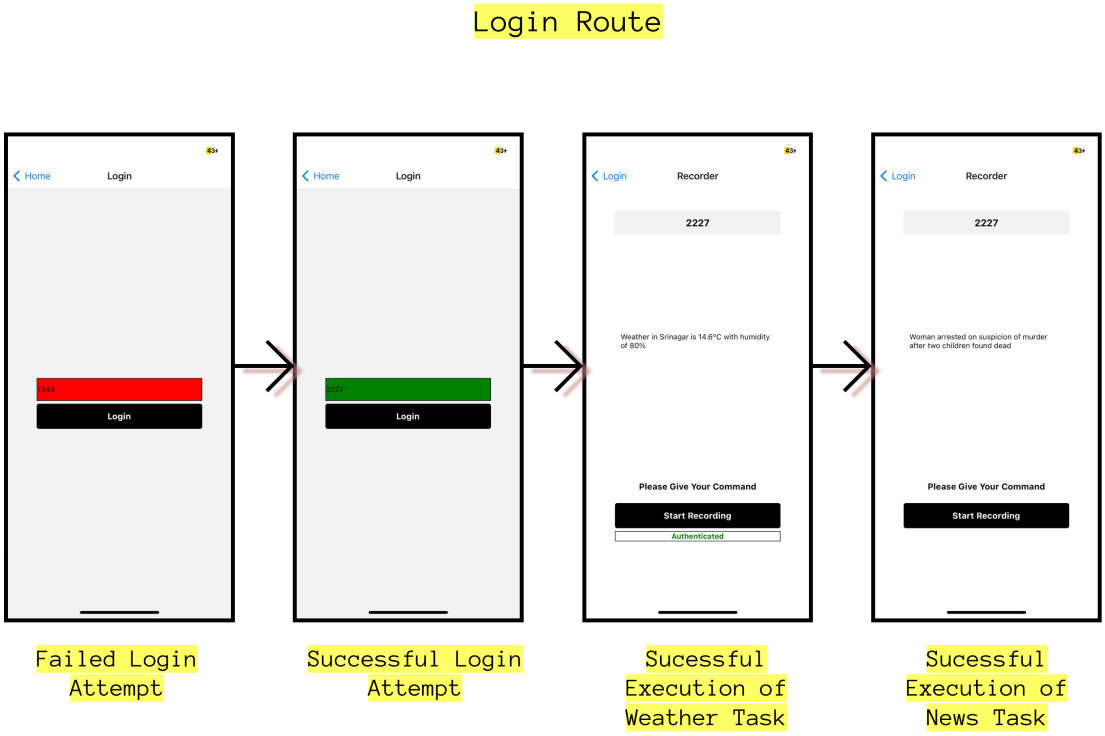

Figure 5.3: Login Route Response of Queries

The above diagram consists of the login route(1-2). In this section the user who have already registered can login with their unique ID. If user is found in the database they are given the green light to go ahead and send commands to the back-end. Images 3-4 consists of successful execution of the weather and the news task. Once the user have said their command , this is send to back-end, where processing is done to identify if the user is authorized or not . Only if authorized their command is executed and result is returned back to the front end of the app.

Chapter 6

Conclusion & Future Work

Personalized voice assistants are becoming more and more important in today's world as they can improve productivity, convenience and accessibility. Personalized voice assistants recognize and adapt to individual user preferences to provide contextual and customized experiences. We can provide personalized recommendations, reminders, and notifications based on your interests, location, and past interactions. This level of personalization not only saves time, it also increases user engagement and satisfaction. Additionally, personalized voice assistants have the potential to support people with disabilities and make technology more inclusive and accessible to all. Advances in natural language processing and machine learning have evolved these assistants to provide more accurate and intuitive interactions. The integration of personalized voice assistants into a variety of devices and services has changed the way we perform tasks, access information and control our surroundings, making voice assistants an integral part of our daily lives. The missing part is now. The objectives that have actually been achieved by the project:

- The presented solution is successfully distinguishing between 251 different speaker with 99.8% accuracy in ideal scenarios i.e noise free clear voice sample of the speaker
- With the help of a pre-trained speech to text like openai/whisper-base we are successfully able to demonstrate flawless speech to text capabilities which along with text classification techniques can partially understand the text and perform the desired function
- With the help of open nature of internet we are able to use these text classification techniques to translate them to skill which can perform the desired function and return the desired result.

There is much scope of further advancements in the project developed, some of which include:

- This model can be incorporated in mobile app in order to provide user a secure way to use the power of their voice. Proof of Concept of the same is presented to you in the form of our app
- More existing voice assistants can use this novel technique to improve their user experiences and make their eco-system more secure

Bibliography

[1] Zhao, Z., Pan, D., Peng, J. & Gu, R. Probing Deep Speaker Embeddings for Speaker-related Tasks. (2022)

[2] Zhang, C. & Koishida, K. End-to-End Text-Independent Speaker Verification with Triplet Loss on Short Utterances. (2017,8)

[3] Pelecanos, J., Wang, Q. & Moreno, I. Dr-Vectors: Decision Residual Networks and an Improved Loss for Speaker Recognition. (2022)

[4] Wang, Q., Okabe, K., Lee, K., Yamamoto, H. & Koshinaka, T. Attention Mechanism in Speaker Recognition: What Does It Learn in Deep Speaker Embedding?. (2018)

[5] Librosa Docs. *Librosa documentation.*[online]
`https://librosa.org/doc/latest/index.html`

[6] Neural Network Crash Course. *Basic Neural Network Crash Course* [online]
`https://developers.google.com/machine-learning/`

[7] Tensorflow's Guide. *Tensorflow's Guide for audio recognition* [online]
`https://www.tensorflow.org/tutorials/audio/simple_audio`

[8] React Native. *React Native Docs* [online]
`https://reactnative.dev/`

[9] Expo CLI. *Expo Docs* [online]
`https://docs.expo.dev/`

Appendices

A. App's Logic which does user authentication and return with results

```
identification(uid, label):
    print("Checking")
    util = utils()
    util.userid = uid
    util.label = label
    files = os.listdir(util.userid + "/")
    data = pd.DataFrame(files)
    data = data.rename(columns={0: "file"})
    data = data[data["file"].str.contains("cmd")]
    data["label"] = util.label
    model = keras.models.load_model("spktvrfidev")
    features_label = data.apply(util.extract_features, axis=1)
    index = data.index[data["file"].str.contains("cmd")].tolist()[0]
    features = []
    features.append(
        np.concatenate(
            (
                features_label[index][0],features_label[index][1],
                features_label[index][2],features_label[index][3],
                features_label[index][4],
            ),
            axis=0,
        )
    )
    X = np.array(features)
    predict_x = model.predict(X)
    preds = np.argmax(predict_x, axis=1)
```

```
    predClass = np.bincount(preds).argmax()
    conn = create_connection(database)
    result = querry(conn, f"select * from user where id='{uid}'")[0]
    if int(result[2]) == int(predClass.item()):
        newfile = convert_to_wav(uid + "/" + str(data["file"][index]))
        text = stt(newfile)
        label = classify_text(text)
        op = ""
        if label[0] == "weather":
            print("Fetching Weather")
            op = get_weather_by_name("Srinagar")
        elif label[0] == "joke":
            print("Fetching Joke")
            op = joke()
        elif label[0] == "fact":
            print("Fetching fact")
            op = fact()
        elif label[0] == "news":
            op = get_news_briefing()
        return True, op
    else:
        return (
            False,
            "You are not authorized to give any commands on behalf of this
                user",
        )
```

B. Text Classification

```
def classify_text(text):
    classifier = pipeline(model="facebook/bart-large-mnli")
    # Perform text classification
    results = classifier(text, candidate_labels=["weather", "joke", "fact",
        "news"])
    # Retrieve the label with the highest score
    max_score_idx = max(
        range(len(results["scores"])), key=results["scores"].__getitem__
    )
    label = results["labels"][max_score_idx]
    score = results["scores"][max_score_idx]
    return label, score
```

C. Speech to text using openai/whisper model

```
    def stt(filename):
    device = "cuda:0" if torch.cuda.is_available() else "cpu"
    pipe = pipeline(
        "automatic-speech-recognition",
        model="openai/whisper-base",
        chunk_length_s=30,
        device=device,
    )
    transcription = pipe(filename, batch_size=8)["text"]
    return transcription
```