International Conference - Innovation-2021-Innovation-2021
International Journal of Scientific Research in Computer Science, Engineering and
Information Technology ISSN : 2456-3307 (www.ijsrcseit.com)
Volume 8, Issue 3, May-June-2021

IVA : An Intelligent Virtual Assistant System Implementation using Speech and Speaker Recognition

Vrushali Kolte[1], Kalyani Kasar[1], Samidha Jadhav[1], Sunil Rathod[2]

[1]Students, Department of Computer Engineering, Dr. D. Y. Patil School of Engineering, Pune, Maharashtra, India

[2]Assistant professor, Department of Computer Engineering, Dr. D. Y. Patil School of Engineering, Lohegaon, Pune, Maharashtra, India

ABSTRACT

Currently, many people use IOT-based voice recognition devices such as Siri by Apple, Alexa by Amazon and Echo from Google. Where there is a boost in IOT devices to the contrary, there is hardly any speech recognition software like Cortana that has very few desktop features. Another fall of these technologies is the security issue because it stores its data in the cloud which can be recovered by any technique and can be used improperly. To overcome above issues this paper proposes IVA, a voice-based intelligent virtual assistant comprising of speech recognition and speaker recognition technology specifically for windows operating system. IVA incorporates the ability to recognize the user's voice to check for security violations and also provide a personalized account to the end user. IVA firstly checks if the user is an authorized user or not and if user voice is detected successfully, then it opens user personalized account to perform some of the common tasks such as playing audio/video files, searching through the web, setting alarms and scheduler, etc.

Keywords : IVA, Speech recognition, Speaker verification, pyttsx3, MFCC, GMM, python 3.7.2, pyqt5, Windows OS.

I. INTRODUCTION

The term virtual assistant was coined in the 1950s, even prior to Siri, which was developed by Apple as a virtual assistant for Android. The term virtual assistant or virtual personal assistant is an application program capable of understanding natural human language, speaking natural language and completing an electronic task for the end user [5]. The main aim is to design a voice-based intelligent virtual assistant (IVA) that acts as a digital organizer to provide a variety of services to its master with the use of various machine learning algorithms, which accept voice input, process it and provide the desired output to the user.

This intelligence system is classified into three generations: First Generation based on Pattern Matching; Second Generation including techniques of Artificial Intelligence such as deep neural network; Third Generation indulges higher ordered, sophisticated pattern matching techniques which are mostly based on AIML, a markup language for chatbots constructions which is based on XML [1].

The simplest and quickest way to communicate is through our own voice. Therefore, voice assistants are in great demand these days. Voice assistants are used in a wide range of areas such as chatbot, home automation, web search, map browsing, etc [4]. This method of interaction with technical apparatus makes lexical communication better than typographic communication. Speech recognition is the backbone of the voice assistant, facilitating the interaction between the system and the end user.

Voice recognition can be broken down into two categories: speech recognition and speaker recognition [7]. Speech recognition is the ability of a machine or program to identify words pronounced by the user and convert them into readable text. It works in three stages: speech to text, text to intention and intention to action. Speech recognition involves many fields of physiology, psychology, linguistics, computer science and signal processing, and is even related to the person's body language, and its ultimate goal is to achieve natural language communication between man and machine.

On the other hand, speaker recognition in its current phase is relatively immature and has very few applications. Its use can be seen in areas like surveillance, authentication and medico-legal recognition of speakers. Privacy is the most important aspect of voice assistants [6]. Speaker recognition can be integrated with voice recognition, enabling users to have a personalized experience, reducing the risk of malware. The voice of each individual has been different just like a fingerprint and can be categorized according to timbre, pitch, length of the vocal device, sound frequency, etc.

II. LITEARTURE SURVEY

Below are some of the highlighted researches in speaker recognition and speaker identification field:

TABLE I: LITERATURE SURVEY TABLE

Sr. No.	Paper Name	Advantages	Limitations
1.	Domain Specific Intelligent Personal Assistant with Bilingual Voice Command Processing [2]	-Commands are processed in two languages- English and Bengali. -Language processing performed using finite automata.	-Noise cancelling module cannot be implemented.
2.	Artificial Intelligence-based Voice Assistant [8]	-understand command easily. -Audio/Video files are easy to download.	-Security of user is at risk.
3.	Application of Automatic Speaker Verification Techniques for Forensic Evidence Evaluation [10]	-Interpreted verification result in terms of probability. -The difficulty of matching two vocal samples is eliminated.	-Low accuracy and reliability.

III. PROPOSED SYSTEM

To make our design easier to understand, we have shown the flow of information through the IVA figure-1; here voice input is the starting point of the system and IVA respondent is the end point.

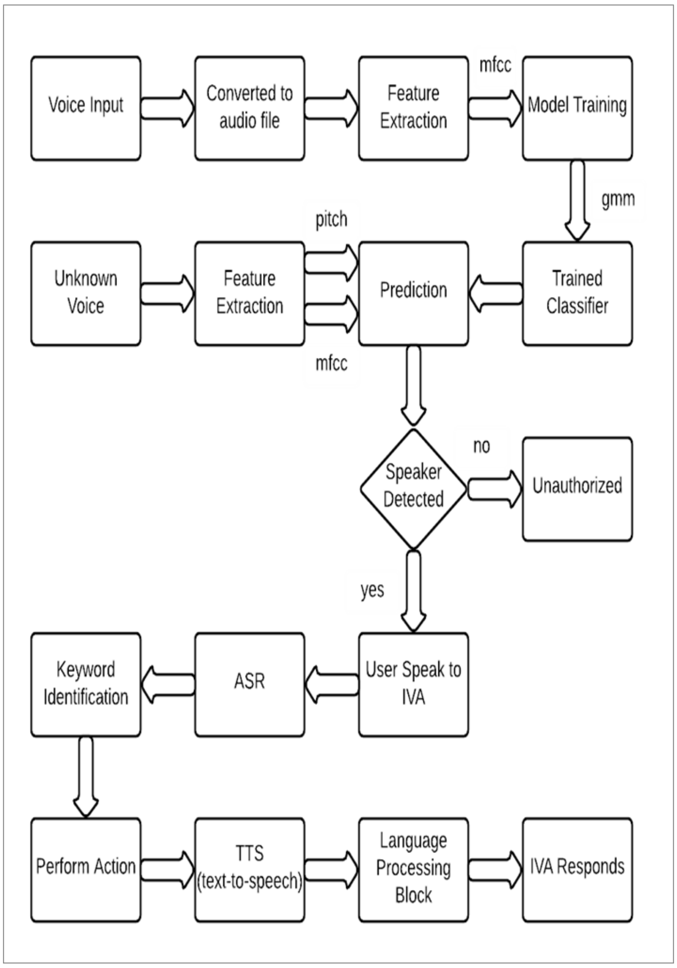


Fig-1: System Architecture of the IVA

A. Audio Processing

Firstly, IVA greets user and ask if he/she is a new or old user. The user speaks into the microphone and if the response is an old user, then the user needs to say the complete phrase specified by the IVA. This audio is captured from the microphone and saved with an extension of .Wav. The recorded sound is subsequently used in the feature extraction phase. If the response is a new user, then the user should record his voice three times saying the same sentence which will generate three audio files. Depending on the recording device, the audio file may have features that cannot be processed through voice recognition engines [2]. Hence, the audio file's properties must first be converted to Mono Channels, 44100Hz Sampling Rate, and 1024 Chunk Size. This standardizes all the audio inputs to the speech recognition engine.

B. Feature Extraction

Mel-Frequency Cepstral Coefficient (MFCC) is the most widely used algorithm for extracting features. Python provides a library known as python_speech_features that provides common features for ASR, including MFCCs and filter bank energies. This module can be used to compute MFCC features such as signal, sample rate, Winstep, numcep, etc. Audio files that are recorded for enrolment or testing go through feature extraction where the frequency of the audio signal is divided into sub-bands using the MEL scale. Then the Cepstral coefficient is extracted from the sub-bands by means of the Discrete Cosine Transform (DCT) [9]. The MEL scale in MFCC is based on the way humans distinguish between frequencies, making audio processing extremely convenient.

MFCC is based on a linear cosine transformation of a logarithmic power spectrum on a nonlinear Mel scale in the frequency range. The Mel scale is linear up to a frequency of 1kHz and then behaves logarithmically. In advancement to the Fast Fourier transform (FFT) parameters which were earlier used for feature extraction, in MFCC, the frequency bands are positioned logarithmically (on Mel scale) rather than being linearly spaced in case of FFT [10]. The MFCC flow chart is depicted in Fig. 2 and Algorithm-1 provides the step-by-step methodology for calculating it.

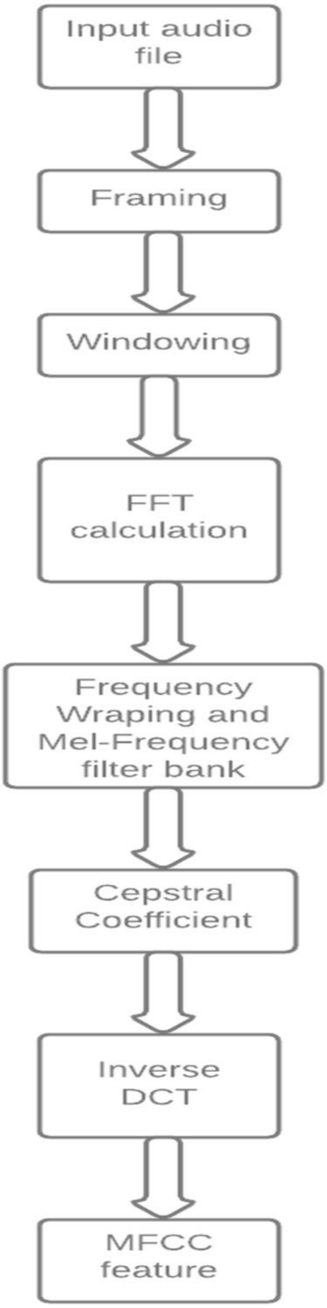


Fig 2: MFCC block diagram

Algorithm-1: Algorithm for MFCC calculation

Steps:

1. Sample speech signal at 16 kHz. Take N=400 (no of samples) and M=0.01 (overlapping factor),
2. Apply 0.025s Hamming window on each frame,
3. Take the 1200-point Discrete Fourier Transform of the frame,
4. Calculate the energy of the filter bank by multiplying each filter bank by the power spectrum and then summing the coefficients,
5. Apply signal on Mel filter bank for MFCC calculation. Convert frequency scale to Mel scale by using below formula, M(f) = 1125 1n(1 + L/700)
6. Apply the Discrete Cosine Transform (DCT) to calculate only the effective portion that is sufficient for the ASR and then the DCT is calculated.
7. We would obtain 12 delta coefficients, which would result in a characteristic vector of length 24.

C. Model Training

We have used the speaker verification method based on suitable Gaussian mixture models (GMM) as the underlying technique, because this mode has a good recognition capability. One of the powerful attributes of the GMM is its ability to form a smooth approximation to the underlying long-term sample distribution of observations obtained from utterances by a given speaker [9]. A GMM is a probability distribution model. When base distributions such as the Gaussian or Cauchy distribution model a single peak, GMMs can model distributions with numerous peaks. This is achieved by adding many Gaussian together. Using a sufficient number of Gaussians, and adjusting their means and covariances as well as weights, almost any continuous density can be approached at arbitrary accuracy.

Python delivers the GMM package. This package is used to estimate parameters for a Gaussian mixture distribution. Parameters used in our system are: n_components: int, covariance_type {“diag”}, n_init: int. The n_components specify the number of mixture components, covariance_type specifies type of covariance parameters to use and n_init specifies the number of initializations to perform. Hence the gmm files are generated in the modelling phase. During the identification or verification stage, the extracted characteristics are compared with the models stored in the speaker database. On the basis of these comparisons, the user is considered to be valid or invalid.

Algorithm-2: Gaussian mixture models (GMM)

Steps:

```
1.  Import required files such as
    -numpy
    -cPickle
    -GMM from sklearn.mixture.
2.  Now import audio files which contains recorded voice of user.
3.  From imported audio files extract 40 dimensional MFCC and Delta MFCC features.
    Vector = extract_features(audio,sr)
4.  Concatenate features of audio files recorded during enrolment phase.
5.  If counter more than no. of audio files.
6.  Initialize gmm parameters gmm = GMM(n_components, covariance_type, n_init)
7.  Dump trained guassian model using cPickle library, set count=0
8.  Compute log-likelihood function.
9.  Put some convergence criterion
10. If the log-likelihood value converges to some value (Or if all the parameters converge to some values) then stop,
11. Else return to step 6
```

D. Speech Recognition

Once a user is detected, an IVA can execute different user-directed tasks. Python provides a speech recognition package that includes speech_recognition module, which is used for converting speech into text. In our project, we installed the pyttsx3 engine package to make IVA talk like a regular human being. ASR will convert the audio received from microphone into readable text. This text is divided into segments and keywords are identified from these segments. These keywords are matched with the keywords mentioned in the queries and that program is executed finally. The activity performed by the system takes the form of a voice. Here pyttsx3 is used when converting text into speech.

IV. TASK PERFORMED BY IVA

- Enrolls user voice with his/her name successfully.
- Search for any required content on Bing if prompted "Search for..."
- The user's voice can be converted to text and saved in a notepad.
- User can set alarm.
- Daily tasks can be scheduled by letting the system know the time, date and event.
- User can ask for any queries using chatbot feature.
- The system can be started or shut down with a simple user command.
- The system continuously asks for any task to be performed till user give a command like "stop working".

V. FUTURE WORK

This project mainly focuses on "Text-dependent recognition". But we feel that the idea can be extended to "Text-independent recognition" and ultimately create a system where users can directly command to execute a given task and the system is able to recognize whose voice it is and also accomplish the given task. This will increase the robustness of the system. The system could be improved to work satisfactorily in various training and testing environments. Noise is a really big deal in both speaker recognition and speech recognition system. It is therefore advisable to use the noise filtration technique to reduce background noise. In addition, numerous features such as uploading a file to the user cloud, playing games, sending emails, etc. Can be applied to the system for greater reliability.

VI. RESULT

The required packages of the Python programming language have been installed and the code was implemented using Spyder (IDE) and below are the few outputs which we have received in our AI-based voice assistant.

A. Detecting valid or invalid user

As shown in below Fig:2, This is how our system window appears when user prompts command "Open IVA"

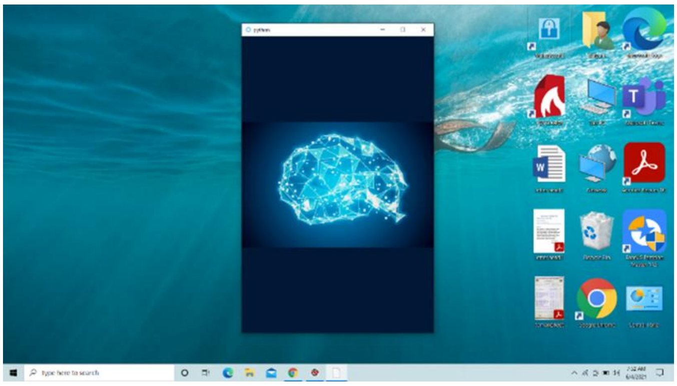


Fig 3: Opening GUI

B. Detected valid user

As shown in below Fig:3, This is how our system window appears when the user is detected as a valid user. From here onwards user can ask the voice assistant to perform any function.

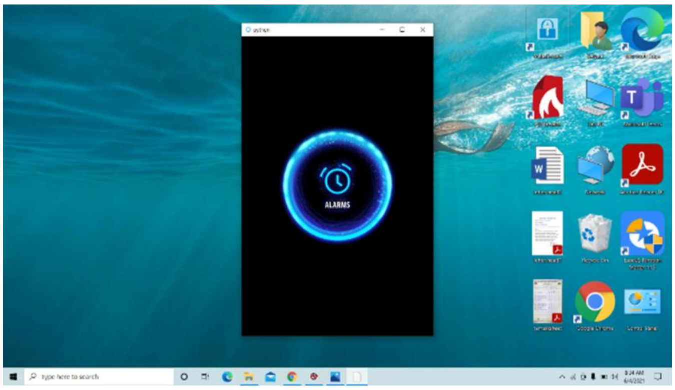
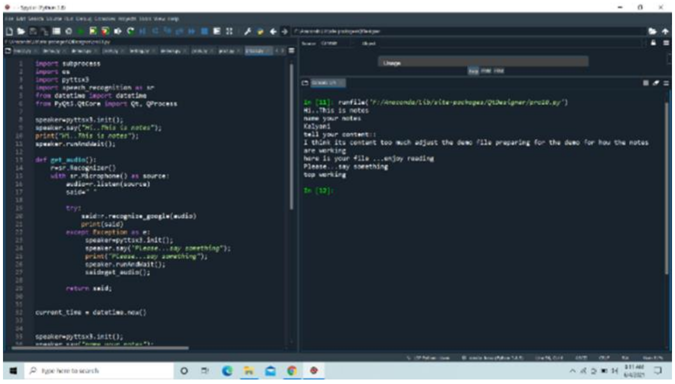


Fig 4: User Detected GUI

C. Creating text documents using notepad

As shown in below Fig:4, When we can ask IVA to "open notepad". IVA ask for document name and content to write in the document.



```
import subprocess
import os
import pyttsx3
import speech_recognition as sr
from datetime import datetime
from PyQt5.QtCore import Qt, QProcess

speaker=pyttsx3.init();
speaker.say("Hi..This is notes");
print("Hi..This is notes");
speaker.runAndWait();

def get_audio():
    r=sr.Recognizer()
    with sr.Microphone() as source:
        audio=r.listen(source)
        said=" "

        try:
            said=r.recognize_google(audio)
            print(said)
        except Exception as e:
            speaker=pyttsx3.init();
            speaker.say("Please...say something");
            print("Please...say something");
            speaker.runAndWait();
            said=get_audio();

        return said;


current_time = datetime.now()


speaker=pyttsx3.init();
```

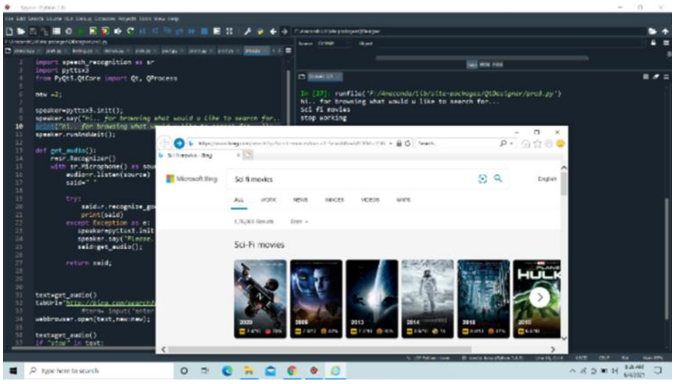
```
In [11]: runfile('F:/Anaconda/Lib/site-packages/QtDesigner/pro10.py')
Hi..This is notes
name your notes
Kalyani
tell your content::
I think its content too much adjust the demo file preparing for the demo for how the notes
are working
here is your file ...enjoy reading
Please...say something
top working

In [12]:
```

Fig 5: Create Notepad

D. Search Output

As shown in below Fig:5, When we ask IVA to "search sci-fi movies", it receives the request and performs the action by searching over Bing.



```
import speech_recognition as sr
import pyttsx3
from PyQt5.QtCore import Qt, QProcess

new =2;

speaker=pyttsx3.init();
speaker.say("Hi.. for browsing what would u like to search for..
print("Hi.. for browsing what
speaker.runAndWait();

def get_audio():
    r=sr.Recognizer()
    with sr.Microphone() as sou
        audio=r.listen(source)
        said=" "

        try:
            said=r.recognize_go
            print(said)
        except Exception as e:
           speaker=pyttsx3.init
           speaker.say("Please.
           said=get_audio();

        return said;



text=get_audio()
tabUrl="http://bing.com/search?
            #term= input("enter
webbrowser.open(text,new=new);

text=get_audio()
if "stop" in text:
```

```
In [27]: runfile('F:/Anaconda/Lib/site-packages/QtDesigner/pro3.py')
hi.. for browsing what would u like to search for...
sci fi movies
stop working
```

Sci fi movies

Sci-Fi movies

Fig 6: Browse

VII. CONCLUSION

This paper introduces voice-based intelligent virtual assistant (IVA) specifically designed for windows operating system. In this system we have integrated both speech recognition and speaker recognition technology. This IVA system uses the voice communication mode to interact with people. The goal of this project was to create an integrated version of both the fields of voice recognition, thus by providing user with personalized access.

During this project, we discovered that the convolutional neural network (CNN) -based approach provides us with independent text verification, which is more preferable than MFCC and GMM. Furthermore, this system can be used in various areas such as home automation, medical assistance, auto automation, robotics and security access, business assistant on the PC [3], etc.

VIII. REFERENCES

[1]. Ravivanshikumar Sangpal, Tanvee Gawand, Sahil Vaykar, and Neha Madhavi, of Computer Technology, Government Polytechnic Pen “JARVIS: An interpretation of AIML with integration of gTTS and Python” 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT).

[2]. Saadman Shahid Chowdury, Atiar Talukdar, Ashik Mahmud, Tanzilur Rahman, “Domain specific Intelligent personal assistant with bilingual voice command processing”, IEEE 2018.

[3]. Polyakov EV, Mazhanov MS, AY Voskov, LS Kachalova MV, Polyakov SV “Investigation and development of the intelligent voice assistant for the IOT using machine learning”, Moscow workshop on electronic technologies, 2018.

[4]. Veton Kepuska and Gamal Bohota “Next generation of virtual assistant (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)”, IEEE conference, 2018.

[5]. Virtual assistant: What is it, 10 2017 [online] Available: www.searchcustomerexperiences.techtarget.com

[6]. Laura BURbach, Patrick Halbach, Nils Plettenberg, Johannes Nakyama, Matrina Ziefle, Andre Calero Valdez “Ok google, Hey Siri, Alexa. Acceptance relevant of virtual voice assistantS”, International communication conference, IEEE 2019.

[7]. Satyam P. Todkar, Snehal S. Babar, Rudrendra U. Ambike, Prasad B. Suryakar Department of Computer Engineering Sinhgad College of Engineering Pune, India “Speaker Recognition Techniques: A review” 2018 3rd International Conference for Convergence in Technology (I2CT), Apr 06-08, 2018.

[8]. Subhash S, Prajwal N, Siddhesh S, Ullas A, Santosh B Department of Telecommunication Engineering Dayananda Sagar College of Engineering Bengaluru, India “Artificial Intelligence-based Voice Assistant”, 2020 IEEE.

[9]. Shilpa S. Jagtap and D.G.Bhalke Department of Electronics and Telecommunication Engineering Rajarshi Shahu College of Engineering, Tathawade, Savitribai Phule Pune University, Pune, India “Speaker Verification Using Gaussian Mixture Model”, 2015 IEEE.

[10]. A.M.T.S.B. Adikari, S. Devadithya, A.R.S.T. Bandara, K.C.J. Dharmawardane and K. C. B. Wavegedara Department of Electronic and Telecommunication Engineering University of Moratuwa Moratuwa 10400, Sri Lanka “Application of Automatic Speaker Verification Techniques for Forensic Evidence Evaluation”, IEEE.