

# Low cost language recognition system

Prachi Pise<sup>1</sup>, Prof Sunita Deshmukh<sup>2</sup>

PG Student, SKNCOE, Vadgoan, Maharashtra, India<sup>1</sup>

Assistant Professor, SKNCOE, Vadgoan, Maharashtra, India<sup>2</sup>

**Abstract:** This paper presents a brief review of speaker & language recognition system using Hidden Markov Model (HMM). For accurate personal identification systems the use of biometric is preferred over the security system implemented by password and pin number. Speech recognition was biometric feature of voice of the speaker. Different speaker have different voice characteristics, these different characteristics are achieved by extracting feature vectors as MFCC from speech. The brief history of the hidden Markov Model explain about voice signal and the evolution of the HMM is done in Google Web API & this is implemented using low cost raspberry Pi. With the help of these techniques the performance has increased. The outputs of system are through speaker.

**Index Terms:** Speech recognition (SR), MFCC, Hidden markov model (HMM), and Raspberry pi, Google Web API, Microsoft Web API.

## I. INTRODUCTION

Speaker recognition is an important branch of speech processing. Speaker recognition is a biometric hearing which is uses for a person's voice for recognition reason. Speaker recognition depends on features of speech signal. Different person has different vocal tract and also different behavioral characteristics. Speaker recognition is non-contact identification system & has used in different application as in the office of judge, military, & information services etc. Language recognition system is a crucial step in various applications such as speech dictionary, IVR system, Voice authentication systems used in security systems. A language model is used in speech recognition systems and automatic translation system to improve performance of system. The function of the system is to recognise the language in which the command is given and then pass it on to the translation system to be used for further use for respective application. Modern speech and language processing is heavily based on common resources: raw speech and text corpora, annotated corpora and tree banks, standard tag sets for labeling pronunciation, part of speech, parses, word-sense, and dialog-level phenomena. Scientific study of language is called linguistics. Estimates number of language in world vary between 5,000 & 7,000. In that, natural language are spoken on signal, but any language is information into code which will be applied through audio i.e. speech, visual.

Language identification applications have two main categories: pre-processing for machine understanding systems and pre-processing for human listeners [8]. One example of the first category is a multi-lingual voice controlled information retrieval system. An example of the second category is the language identification system used to route an incoming tell phone call to a human switchboard operator fluent in the corresponding language. Numerous applications for automatic spoken language identification can be found in day-to-day life. Tourist

agencies may need automatic language identification systems to quickly find out what language a customer prefers and provide more efficient services. Due to terrorist attacks in recent years, homeland security becomes extremely important in all countries; the automatic language identification technology then can be used to pre-process and filter the suspected speech without being noticed by the terrorists.

By speech and language processing, we have in mind those computational techniques that process spoken and written human language, as language. Language is very important and it will contribute everything from everyday application as word counting automatic hyphenation to automatic question answering on the web and real-time spoken language translation. Difference between language processing and data processing is their use of "knowledge of language". Example: UNIX WC program which is used for count the total number of bytes, words and lines in a text file. Unix WC program used as count bytes and lines. When it is used to count the words in a file it requires knowledge about what it means to be a word, and thus becomes a language processing system. WC is an extremely simple system with an extremely limited and impoverished knowledge of language.

Summarize, the knowledge of language needed to engage in complex language behavior can be separated into six distinct categories.

- Phonetics and Phonology – The study of linguistic sounds.
- Morphology – The study of the meaningful components of words.
- Syntax – The study of the structural relationships between words.
- Semantics – The study of meaning.
- Pragmatics – The study of how language is used to accomplish goals.

- Discourse – The study of linguistic units larger than a single utterance.

## II. RELATED WORK

In [8], Recent progress in microelectronics conducive to acquiring powerful microprocessors which is used to minimize execution time with high computational cost. Because of this progress in field of microelectronics complexity increases, that might result into use of non-natural function or floating-point operation. Progress in biometric speaker recognition algorithm gives attention to improve its performance in terms of FRR (False Rejection rate), FAR (False acceptance Test), or ERR (Equal error rate). Three aspects need to achieve system robustness, space, and cost.

Biometric speaker identification hearing in which samples of voice are getting through low cost sensor device. In this Support vector Machines (SVM) method uses for speaker recognition which gives good recognition rate. Some dealing with FPGA (Field Programmable Gate Array) implementation of generic SVM for classification [4].

In this paper hardware is specific and implementation on SVM speaker verification system. System has various stages for feature vectors based on MFCC (Mel-frequency Cepstral Coefficients) and their related deltas with matching vectors which is stored in SRAM memory.

In [5], In past two decades , informative advancement in ASR (Automatic Speech Recognition) and in (MT) Machine translation. Obvious there are some problem in which many problem in ASR & MT are related to each other. In this lecture, there is explanation about fundamental connection between ASR and MT and gives unified ASR discriminative training paradigm which is recently developed & extended to train MT model. Aim of ASR and MT is to translate speech from one language to another.

In ASR, MT &ST, ASR and MT are considered as subcomponent of ST, in this ASR and MT is tightly related as one unit. So both complexity & capability of such type of system increases & model learning is crucial. ASR is the operation their related translating speech technology into series of words .General purpose speech recognition system uses HMM (Hidden Markov Models) for acoustic modeling and for language recognition N-gram Markov Model. Achievement of ASR in last few years in major four areas:

1. Infrastructure area
2. Area of knowledge representation
3. Area of modeling and algorithms
4. Area of recognition hypothesis

MT operation for converting text in one language that is sources language into other language which is of destination language. Sequences of important operation completed in MT component technology as, word

alignment, phrase-based methods, syntax-based MT methods, discriminative training methods, and system combination method. Today MT is used publically e.g. via Google translation service and Microsoft translation service.

In ST system two signals one of them is sources as input signal which translates to target language as second signal. This in the form of text as well as speech for synthesis. ASR and MT subcomponent of ST. Integration of ASR and MT is in an end-to-end ST model. Integration of ASR & MT is at decoding stage. Small progress on optimal integration for learning of ST system.

In [6], Speaker recognition divided into two, first is speaker identification (finding identity) second is speaker verification (Authenticating claim of identity).A closed set of speaker identification system select speaker in training set who best matches unknown speaker. An open set of speaker identification allows possibility that unknown speaker may not exist in training set, so alternative is required for unknown speaker. Traditionally, speaker identification system running software on single microprocessor and that software operate sequentially so slow for high throughput real time application. This paper describes FPGA implementation for GMM based speaker identification. Improvement in FPGA is DSP which gives high performance & low cost Aim is to achieve a system that can [process large number of voice stream in real time

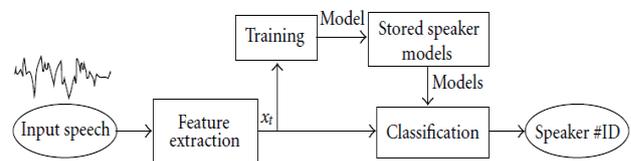


Fig. 1 Top-level structure of speaker identification system

In Figure 1 speaker identification system designed to implement text-dependent speaker identification. Input is in speech signal which sampled and transferred into digital format. A characteristic of speech signal is of feature vector which is extracted from input signal in the form of Mel-Frequency cepstral coefficient (MFCC). System then divided into training and classification. In training, every registered speaker has to give samples of their speech to the system and can train reference model for that specific speaker, while in classification input speech is matched with stored reference model and identification done.

Feature Extraction: Aim of feature extraction is to convert speech waveform to set of feature which will be useful for feather analysis. Short speech is of short time, its characteristics rarely changes while in long speech has long period reflects different sound feature extracted by MFCC, Linear prediction coding (LPC) etc. MFCC chosen as based on perceptual characteristics of human auditory system.

In [2], this explains implementation of embedded speaker verification system used in electronic door lock and other

application. System implement on dsPIC from microchip which is combination of microcontroller and characteristics of DSP in 16-bit high-performance. Objective of system is to achieve maximum efficiency of code for speed & memory storage. Result of system with false acceptance rate of 8% for false rejection rate of 12%. Biometric speaker system has two modes: identification & verification .In identification, objective is find speaker in unknown users which already stored in database. In verification objective is find whether a known user is verified in database.

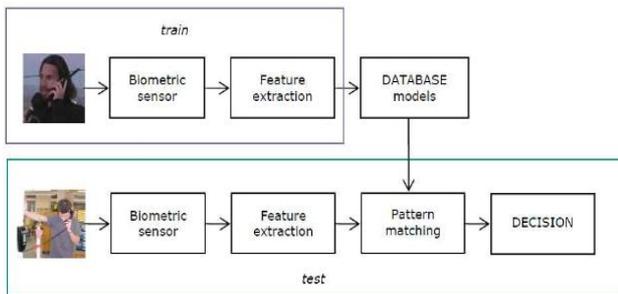


Fig. 2 Architecture of typical biometric recognition system.

This system divided into two training & testing. In training phase through biometric sensor get voice of user. Voice characteristics extracted from feature vector of users voice. In testing through sensor unknown speaker voice is matches with database in pattern matching.

In [1], translation of language require large amount of human knowledge, this knowledge is to be encoded in machine-process able form. So (MT) machine translation is very challenging. Natural language that is spoken having more than one possible meaning. Two language are rarely defines same content in same way. Google translate API (2011, SR, MT, ST-A Unified) is used to built language & translation model from huge amount of texts through statistical techniques. Other tools like IBM Lotus Translations gives cross language services.

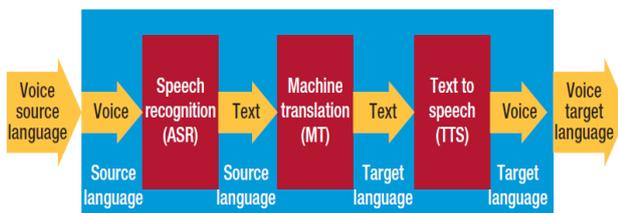


Fig. 3 Speech-to-speech translation

Three components in speech-to-speech translation: (ASR) automatic speech recognition, MT, and voice synthesis (TTS text-to-speech). In Figure 3, operation on sources voice language converted into text whatever source speaker speak all happen in ASR. This sources language as text then goes to machine translation to convert to target language in the text form. End, translated language which is in text then goes to TTS and translated language is in voice form that is voice target language.

### A .Recent Technology in Speech Recognition

Parts of program on speech-to-speech translation here see in table 1, [2014]. Some recent technology available for speech recognition.

Microsoft speech API which is for Windows. This API supports small number language as American English, British English, Spanish, French, German, simplified Chinese, and traditional Chinese. As these are native windows API. For this you should be good enough for C++ developer. [10].

Microsoft.NET is another way to achieve windows speech recognition through system, for this C# developer is needed.

Microsoft Sever-Related Technologies gives access to speech & their synthesis to develop complex voice/telephony application. This technology supports 26 different languages. [11].

Sphinx 4 new open sources speech recognition is based on hidden markov models (HMM) & develops on JAVA programming language.This allows implementation of continuous-speech, speaker-independent, and large-vocabulary recognition systems [12].

HTK also referred as HMM Toolkit, this toolkit used as building & manipulating HMMs.HTK also used in speech synthesis, character recognition, and DNA sequencing. HTK can form complete continuous-speech, speaker-independent and large-vocabulary recognition systems for any demanded language. It also supplies tools to create and train acoustic models [13].

Julius gives high-performance decoder & support large vocabulary and continuous speech recognition which is very important features for dictating systemic .Job of decoder is to identify most likely spoken words for given acoustic prove. In this system different models are used who is created in different tools as HTK and the Cambridge Statistical Language Modeling toolkit (CAMSLM) from Carnegie Mellon University (CMU; [14].

Julius is freely available & this will work on mainly Linux but also works on windows & Microsoft SAPI-compatible version.

Java Speech API is used for cross-platform APIs which is supports to command-and-control recognizers, dictation systems and speech synthesizers. Recently java speech APL includes yjavadoc style API which has 70 classes in and interfaces to API (www. oracle. Com /technetwork/java/jsapifaq-135248.html). Advanced speech application programming with JSAPI & this JSAPI is freely available.

Google Web Speech API, era of 2013 Google released chrome version 25which is support speech recognition of

different language through Web Speech API. This support to JavaScript library that developer integrates continuous speech recognition features as voice in Web application. This is allowed to chrome browsers only. Other browsers don't support JavaScript library [15].

Nuance Dragon SDK support several language as French German, Italian, and Dutch. This applied to Desktop, PC and MAC and MAC as well in mobile app for Android & iOS .Nuance also gives software development kit for development of window application SDK as backend and mobile SDK develop apps for iOS, Android, and windows phone.

### III. PROPOSED SYSTEM

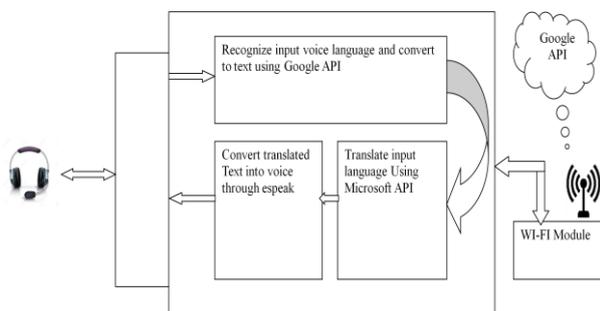


Fig. 4 Block Diagram of proposed system

#### A. Raspberry PI

Implementation of speech and language recognition system is done on Raspberry pi. Raspberry pi is low cost controller which is based on ARM processor.

Raspberry Pi [16] is low cost computer. Since from 2012 raspberry has great success achieved, as it has large part together to its small size with low power consumption, all interfaces, and peripherals. Mostly it is run on Linux operating system which is good for embedded system. Raspberry Pi is used for speaker verification system. Difference is that system clock has to be reconfigured to operate at different frequency so that system will run faster. In Raspberry Pi audio files is recorded using a USB microphone.

#### B. Components of the system

1. Mic, at start it behave as input.
2. Raspberry Pi, handles communication between Google API and User.
3. Google API for speech recognition.
4. Encode translated data into sound using espeak.

#### C. Raspberry Pi Communication

Raspberry PI is the low cost minicomputer which is interfaced with internet and allows it to communicate with the World Wide Web. RP handles all communication with user through headphone and internet (Google API).

#### D. Language Recognition

For language recognition we use Google API which online recognize language and convert into other language. For

translation from one language to other we use Microsoft API. For language recognition it uses its own database and using HMM algorithm recognize the word. The recognize word is then convert to other language. Methodology used here is all take care by Google API as well as Microsoft API.

#### E. Encoding of Text into Sounds

The translated word or series of sentence is then converted into voice through espeak, a command line program (Linux and Windows) to speak text from file.

#### F. Stepwise execution of project

1. User gives voice input from mic to Raspberry Pi.
2. Voice input language is recognize by Google API which in turn translate into other language.
3. The translated text language is then encoded into sound in espeak in Linux, a command line program (Linux and Windows) to speak text from a file.
4. The sound in espeak is then transfer as output to headphone.

#### G. The interfacing of headphone/mic and Raspberry pi (RP) required setup shown as follows:

1. Connect USB headphone to RP
2. Check it is install or not also check it is default headphone or not

```
#cat /proc/asound/cards
#cat /proc/asound/modules
```

The above command shows the headphone details that you have attached. If not then go for next step.

3. Enter the following command, it will open the file.

```
#sudo nano /etc/modprobe.d/alsa-base.conf
```

In this file just change the following line

```
Options snd-usb-audio index=-2
```

To

```
Options snd-usb-audio index=0
```

4. Now restart RP and again check the step 2 to check the headphone is install or not

5. Now check the interfacing by do some recording

```
#arecord -d 5 -r 48000 record_sample.wav
```

6. Now check the interfacing by do some recording

```
#arecord -d 5 -r 48000 record_sample.wav
```

7. Check sound recorded or not using below command

```
#aplay record_sample.wav
```

### IV. SETTING OF RASPBERRY PI

#### A. Booting process of raspberry pi

1. When the Raspberry Pi is first turned on, the ARM core is off, and the GPU core is on. At this point the SDRAM is disabled.
2. The GPU starts executing the first stage boot loader, which is stored in ROM on the SoC. The first stage boot loader reads the SD card, and loads the second stage boot loader (bootcode.bin) into the L2 cache, and runs it.
3. Bootcode.bin enables SDRAM, and reads the third stage boot loader (loader.bin) from the SD card into RAM, and runs it.

4. Loader.bin reads the GPU firmware (start.elf).
5. start.elf reads config.txt, cmdline.txt and kernel.img

loader.bin doesn't do much. It can handle .elf files, and so is needed to load start.elf at the top of memory (ARM uses SDRAM from address zero). There is a plan to add elf loading support to bootcode.bin, which would make loader.bin unnecessary, but it's a low priority (I guess it might save you 100ms on boot).

**B. Raspberry pi interfacing**

1. Insert memory card in PC
2. Load Raspbian OS in memory card using Win32DiskImager-0.9.5-binary

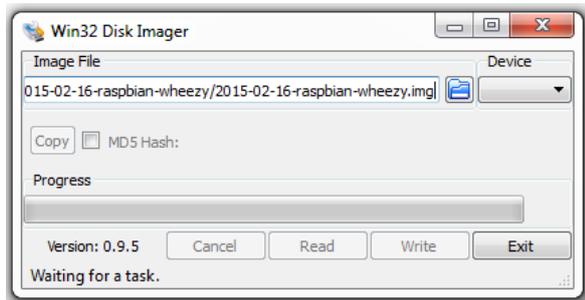


Fig. 5 Win 32Disk Manger.

3. Click on "Write" to load OS in memory card
4. Remove memory card from PC and then insert in Raspberry Pi
- 5.



Fig.6 SD-Card in Raspberry PI Board.

6. Connect power supply to raspberry pi and connect raspberry pi and pc by Ethernet.

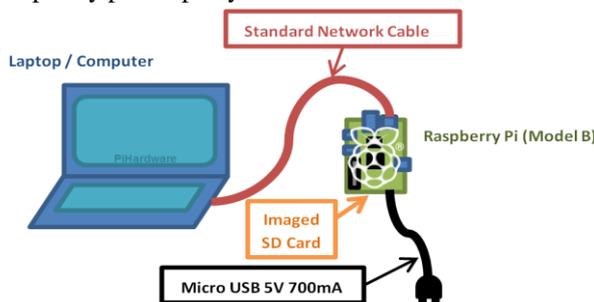


Fig. 7 Connect Raspberry Pi and PC.

7. Now in PC open command prompt and type "ipconfig" to get LAN IP (consider ip = 169.254.148.54). Before share your Wi-Fi internet to LAN.

8. Now remove MMC from Raspberry Pi and insert to PC
9. Open "cmdline.txt" from MMC and type at last "ip=169.254.148.57::169.254.148.54"
10. Now remove MMC from PC and insert to Raspberry Pi
11. After booting of Pi, open Remote Desktop Connection, type 169.254.148.57 in Computer field



Fig. 7 Remote Desktop connection.

12. Click connect -> yes
13. Now window will open in which type username: pi and password: raspberry



Fig.8 Window after remote Desktop connection.

14. Click OK
15. Now you will get Raspberry Pi window on your laptop screen.

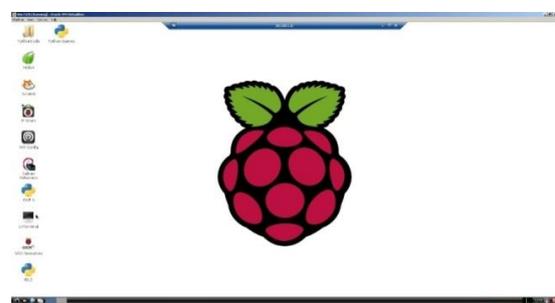
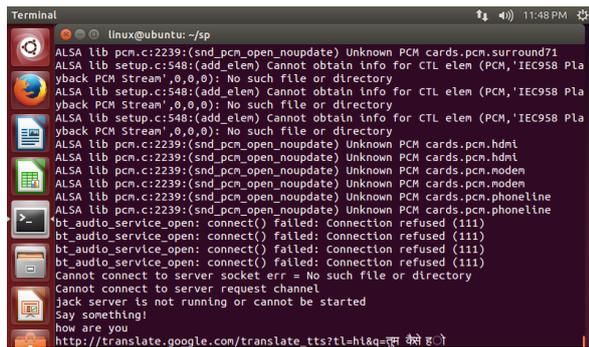


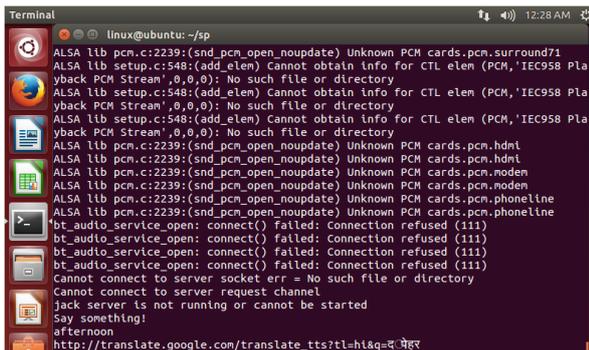
Fig.9 Raspberry pi Window.

**V. EXPERIMENTATION**

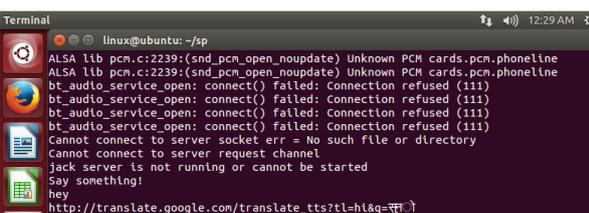
Aim of our system is to convert speech to text through Google API, we can test the input as a speech and get output as text with processing time of speech to text conversion simulating the estimated result before actually implementing it to the hardware.



```
Terminal
linux@ubuntu: ~/$
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.surround71
ALSA lib setup.c:548:(add_elem) Cannot obtain info for CTL elem (PCM,'IEC958 Pla
yback PCM Stream','0,0,0): No such file or directory
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.hdmi
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.hdmi
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneLine
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneLine
bt_audio_service_open: connect() failed: Connection refused (111)
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
Say something!
how are you
http://translate.google.com/translate_tts?tl=hi&q=कैसे हूँ
```



```
Terminal
linux@ubuntu: ~/$
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.surround71
ALSA lib setup.c:548:(add_elem) Cannot obtain info for CTL elem (PCM,'IEC958 Pla
yback PCM Stream','0,0,0): No such file or directory
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.hdmi
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.hdmi
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneLine
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneLine
bt_audio_service_open: connect() failed: Connection refused (111)
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
Say something!
afternoon
http://translate.google.com/translate_tts?tl=hi&q=दूध
```



```
Terminal
linux@ubuntu: ~/$
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneLine
ALSA lib pcm.c:2239:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneLine
bt_audio_service_open: connect() failed: Connection refused (111)
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
Say something!
hey
http://translate.google.com/translate_tts?tl=hi&q=सूतो
```

The speech used to simulate the program in linux through python language. In google API, input speech is taken as input to google API system and after implementing algorithm will get result as text of our speech.

## VI. CONCLUSION

The speaker recognition is thus successfully implemented with an average accuracy. The recognition system primarily uses the HMM or technique to optimize the sentence and do the grammar construction. The Google database is used to analyse the word construction ensuring the accuracy of the word. Google also provides free distribution, control and access the system through Speaker/Mic devices.

The next task is to convert the recognized speech in text and translate to specific language. The complete system will be implemented using Raspberry Pi. There are number of free API's available to recognize and translate speech that can be implemented in project stage-II. Based on the best result the final API will get selected. The result should have good accuracy and fast response time.

## ACKNOWLEDGMENT

I would like to express my sincere thanks to our Head of Department **Dr. S. K. Shah** for her valuable references

and support throughout the seminar work. I thank **Prof. R.H.Jagdale** for her support, co-operation and valuable suggestions. I am grateful to my Principal, **Dr. A.V. Deshpande** and Vice Principal, **Dr. K.R. Borole** for their encouragement and guidance throughout the course. I express my sincere thanks to all teaching and non teaching staff of Electronics and Telecommunication department of Smt. Kashibai Navale College of Engineering- Pune, for the help without which this work was not possible.

## REFERENCES

- [1] Tiago Duarte, Rafael prikladnicki, fabio calefato, and Filippo Ianubile, "Speech Recognition for voice-based machine translation," IEEE computer society, vol. 14, no. 1, pp. 0740-7459, Feb. 2014.
- [2] M. Lizondo, P. D. Agüero, A. J. Uriz, J. C. Tulli and E. L. Gonzalez. 2012. "Embedded speaker verification in low cost microcontroller", Congreso Argentino de Sistemas Embebidos 2012. Buenos Aires, Argentina. 15-17 Agosto, 2012.
- [3] J. Li, D. An, L. Lang, and D. Yang. 2012. "Embedded Speaker Recognition System Design and Implementation Based on FPGA", Procedia Engineering 29, pp. 2633 - 2637, 2012.
- [4] Z. Nie, X. Zhang, and Z. Yang. 2012. "An FPGA Implementation of Multi-Class Support Vector Machine Classifier Based on Posterior Probability", Int. Proc. of Computer Science and Information Technology, vol 53(2), pp. 296 - 302, October 2012.
- [5] Xiaodong he and Li Deng, "Speech recognition, machine translation, and speech translation-A unified Discriminative learning paradigm," IEEE signal processing magazine. no. 8, pp. 126-133, September 2011.
- [6] P. Ehkan, T. Allen, and S. F. Quigley. 2011. "FPGA Implementation for GMM-Based Speaker Identification", International Journal of Reconfigurable Computing, volume 8 pp. 2011.
- [7] Yong Guan, Lin Zheng, Jilei Tian, "Real-time Speaker Adapted Speech to Speech Translation System in mobile Environment", IEEE Vol 10, pp. 4244- 5900, 2010.
- [8] R. Ramos-Lara, M. López-García, E. Cantó-Navarro, and L. Puente-Rodriguez, "SVM Speaker Verification System Based on a low-cost FPGA", International Conference on Field Programmable Logic And Applications, pp. 582 - 586, 2009.
- [9] Hatch, A.O.; Stolcke, A. 2006. "Generalized Linear Kernels for One- Versus-All Classification: Application to Speaker Recognition," Acoustics, Speech and Signal Processing, 2006. ICASSP 2006.
- [10] [http://msdn.microsoft.com/enus/library/hh323805\(v=office.14\).aspx](http://msdn.microsoft.com/enus/library/hh323805(v=office.14).aspx)
- [11] [http://msdn.microsoft.com/enus/library/hh361571\(v=office.14\).aspx](http://msdn.microsoft.com/enus/library/hh361571(v=office.14).aspx)
- [12] <http://cmusphinx.sourceforge.net/sphinx4>
- [13] <http://htk.eng.cam.ac.uk/docs/faq.shtml>
- [14] [http://julius.sourceforge.jp/en\\_index.php](http://julius.sourceforge.jp/en_index.php)
- [15] <http://chrome.blogspot.com.br/2013/02/bringing-voice-recognition-to-web.html>
- [16] <http://www.raspberrypi.org/>
- [17] <http://translate.google.com>
- [18] <http://translator.bing.com>