Architecture of Speech-based registration system

Mayank Kumar, D B Mahesh Kumar, Ashwin S Kumar, N K Srinath

Abstract**—**In this era of technology, fueled by the pervasive usage of the internet, security is a prime concern. The number of new attacks by the so-called "bots", which are automated programs, is increasing at an alarming rate. They are most likely to attack online registration systems. Technology, called "CAPTCHA" (**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part) do exist, which can differentiate between automated programs and humans and prevent replay attacks. Traditionally CAPTCHA's have been implemented with the challenge involved in recognizing textual images and reproducing the same. We propose an approach where the visual challenge has to be read out from which randomly selected keywords are used to verify the correctness of spoken text and in turn detect the presence of human. This is supplemented with a speaker recognition system which can identify the speaker also. Thus, this framework fulfills both the objectives – it can determine whether the user is a human or not and if it is a human, it can verify its identity.

Keywords**—**CAPTCHA, Automatic Speech Recognition, Keyword Spotting.

I. Introduction

THE prime concern for the internet in today's world is security. The number of new attacks being introduced is increasing at an alarming rate. The new attacks have proven to be far more destructive from the earlier ones. One such attack is done by the so-called "bots", which are automated programs. They are most likely to attack online registration systems, wherein the user is asked to enter a username and a password, or online banking systems, where a user is asked to enter a secret PIN; basically any online transaction which entails the user to enter some important details which have to be kept confidential, Technology, like "CAPTCHA" (**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part) do exist, which differentiate between automated programs and humans. In most cases, they are images containing distorted text, which can be comprehended only by humans [1]. These have been successful to quite a remarkable extent. But of late, even they have been vulnerable to attacks and moreover, they cannot differentiate between humans. There arises a need for a more secure system, which can differentiate between automated programs and humans, in addition to differentiating between humans themselves.

An alternative to this is the use of speaker recognition technology. Speaker recognition offers the additional boon of recognizing individuals. Instead of typing the content of the visual challenge, randomly selected keywords in the spoken utterance is used to ascertain the similarity of the displayed text and spoken text. Thus, CAPTCHA technology, when integrated with speaker recognition, fulfills both the objectives of determining whether the user is a human or not as well as recognize who the user is. Thus, if such a security measure can be deployed on the online transaction systems, they can provide double protection due to which the system will be more immune to attacks.

Our approach is an integration of 2 technologies - a CAPTCHA, which has been developed in Visual C++ using Microsoft Foundation Classes (MFC)[2] and Graphics Design Interface Plus (GDI+) library, followed by Automatic Speech Recognition (ASR), which is used to build a "keyword spotting" system for continuous speech, besides the voice recorder module used for recording the input speech.

CAPTCHA and word spotting technology

- *CAPTCHA*

A *CAPTCHA* is a type of challenge response test used in computing to determine that the response is not generated by a computer. The process involves one computer asking a user to complete a simple test which the computer is able to generate and grade. Because other computers are unable to solve the CAPTCHA, any user entering a correct solution is presumed to be human. A CAPTCHA is sometimes described as a reverse Turing test, because it is administered by a machine and targeted to a human, in contrast to the standard Turing test that is typically administered by a human and targeted to a machine.

Automated tests which distinguish humans from computers for the purpose of controlling access to web services were first suggested in 1996 by Moni Naor. [3]

Primitive CAPTCHAs seem to have been developed in 1997 at AltaVista by Andrei Broder and his colleagues to prevent

Manuscript received May 30, 2008.

Ashwin S Kumar is with the Department of Information Science, R V College of Engineering, Bangalore. (Phone: 91-80-23217481; e-mail: ashwinskumar@gmail.com).

D B Mahesh Kumar is with the Department of Information Science, R V College of Engineering, Bangalore. (Phone: 91-09341616713; e-mail: rvce.mahesh@gmail.com).

Mayank Kumar is with the Department of Information Science, R V College of Engineering, Bangalore. (Phone: 91-09986155825; e-mail: mayankiaf@gmail.com).

Prof N K Srinath is the head of the department of Information Science, R V College of Engineering, Bangalore. (Phone: 91-09845293550; e-mail: srinath_nk@yahoo.com).

bots from adding URLs to their search engine. In order to make the images resistant to OCR (Optical Character Recognition), the team simulated situations that scanner manuals claimed resulted in bad OCR. In 2000, Luis von Ahn and Manuel Blum developed and publicized the notion of a CAPTCHA, which included any program that can distinguish humans from computers

As on date most of the CAPTCHA implementations use visual challenges, some of them are claimed to have been already broken. Audio CAPTCHA are also in use for the assistance of the physically challenged persons particularly those who are blind or their inability to use keyboard because of some physical disability. In some sense the proposed approach, which requires the user to read out the contents of a carefully designed a textual image hopes to address both the issues.

The security of any CAPTCHA system stands on a key principle. The algorithm to generate the challenge (the distorted image or the set of questions to be answered) is difficult to break within the stipulated time. This translates to the difficulty in recognizing the text from the given image particularly when the background of the image is such that segmentation is extremely difficult [4]. Added to this is the keyword, selection process that differs from session to session even if the same text is displayed. The dual mechanisms ensure the enhanced security of the proposed CAPTCHA.

- *Keyword spotting in continuous speech*

The detection of keywords requires keyword spotting in continuous speech, which is a particular application of ASR. Here the speech recognizer has to pick out the occurrences of these keywords from the unknown speech. In this case it is important to have some acoustic modeling of non-keywords in order to avoid a large number of incorrect keyword hypotheses (termed as false alarm). The non keyword model component of a word spotter can range from the single acoustic garbage model intended to match all non keyword speech to a large number of models for a vocabulary on non-keywords.

The structure of the simple HMM (Hidden Markov Model) based word spotter is shown in fig 1. The HMM's are arranged in a network. Each keyword is modeled as an individual HMM. A path through the network for implementing the non-keyword speech is also incorporated. The identification of keywords is done by putting a threshold on likelihood of the keyword occurring in a utterance given the target occurrence.

Data flow is divided into six processes as follows:

1. Creation of databases
   - Computation of the average duration of the basic phonetic units which covers spoken words in English language.
   - Create a phonetic dictionary
   - Create a pool of sentences to be used for display in the CAPTCHA text.
2. Generation of Session Parameters
   - Selection of the sentences to be displayed, from the database
   - For each sentence, randomly select:
     - Number of keywords to be used in the verification module
     - The keywords themselves

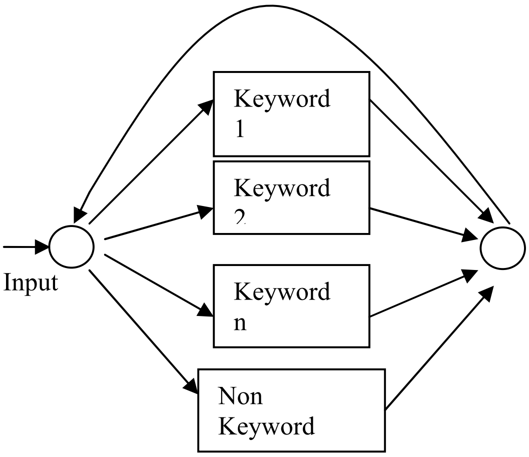


Fig. 1 Representation of Simple HMM based keyword spotting

II. SYSTEM ARCHITECTURE

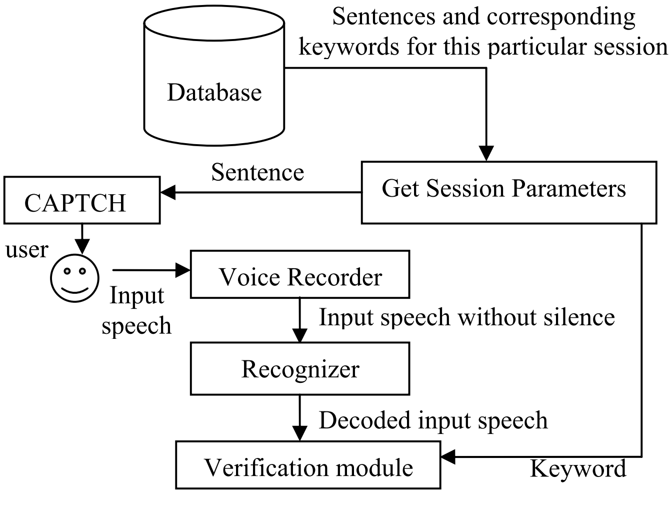


Fig. 2 System Architecture

     - Calculation of the approximate average duration of the displayed sentence in the CAPTCHA by summing the durations of the phonetic units comprising the sentence. (to be used as benchmark in the verification module)

3. Generation and Display of Captcha text:
   - Sentence to image string conversion
   - Boundary constraining and skew

constraining

- Generation of background (of the displayed text) whose color, texture and sub-regions vary with each session.
- Curve generation on the displayed image string

4. Recording the input speech
    - Recording the continuous input speech of the user
    - Removal of periods of silence from the input speech

5. Recognition
    - Speech Parameterization (extracting MFCC parameters from the wav files)
    - Generation of Grammar Network
    - Decoding of the input speech

6. Verification
    - Ordering of Keywords
    - Keyword comparison
    - Announcement of binary decision of allowance or denial

III. DETAILED DESIGN

Database creation

- **Create database of phonetic units:** Based on phonetic study of the English language, British English has forty six phonetic units. A list of these phonetic units is created containing entries in the following fashion

  [PHONETIC UNIT] [AVERAGE_DURATION]

  The average duration't' is calculated by computing the mean duration of the phonetic unit, from the available phonetic transcriptions a large number of voice samples.

  The average duration of the phonetic units is used to determine the approximate duration of the displayed sentence which is to be used as a benchmark in the verification module.

- **Create a dictionary database:** A database containing the words of the English language is created. Each row of the database contains entries in the following fashion

  [WORD] [PHONETIC UNITS....] [SP]

  Each word ends with a short pause (SP).

  The dictionary is created manually by studying the phonetic structure of the English words and the phonetic units which comprise them. For our work we used BEEP (British English pronunciations), which is a pronunciation dictionary made available by Cambridge University consisting of 250000 English words.

- **Create a database of sentences:** A database of sentences and their corresponding keywords has to be created and maintained. Sentences from the database will be chosen and passed to the CAPTCHA module. This is achieved by using the structure shown below:

  {[SENTENCE] [NUM-KEYS] [K1][K2]...[Kn]}

  The sentences present in the database should consist of words from dictionary database only. The sentences are chosen randomly with the constraint that for a particular session the selected sentences should be unique. No repetition of sentences is allowed for a single session.

❖ *Generation of Session Parameters*

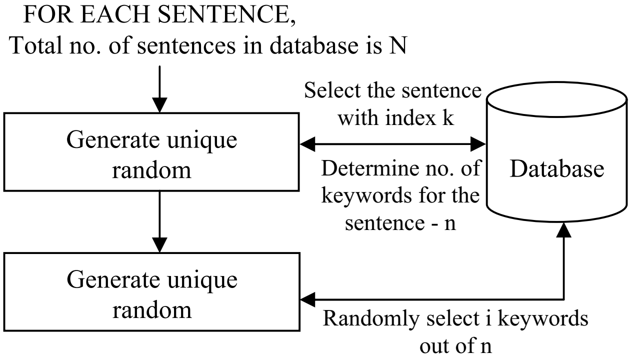

FOR EACH SENTENCE,
Total no. of sentences in database is N
Generate unique random
Select the sentence with index k
Database
Determine no. of keywords for the sentence - n
Generate unique random
Randomly select i keywords out of n

Fig. 3 Generation of Session Parameters

- **Selection of the sentences to be displayed, from the database:** For a particular session, a fixed number of indexes of unique sentences of the sentence database are randomly generated.

- **For each sentence, randomly select**
    - ➢ Number of Keywords

      The total number of keywords (n) for the particular sentence is determined from the sentence database, and a unique random number from 1 to n is selected for the particular session.

    - ➢ The keywords themselves

      After having selected the number of keywords (NUM_KEYS) for each sentence for the particular session, NUM_KEYS numbers of keywords are randomly selected from the sentence database.

❖ *Generation & Display of CAPTCHA text*

- **Boundary constraining and skew constraining:** The bounding box is the one inside which the CAPTCHA image string is displayed. To enhance the complexity of the image, skewness factor is introduced by stretching the four corners of bounding box. Care is also taken to ensure that the readability is not affected. Moreover, it is made sure that the image always lies within the bounding box.

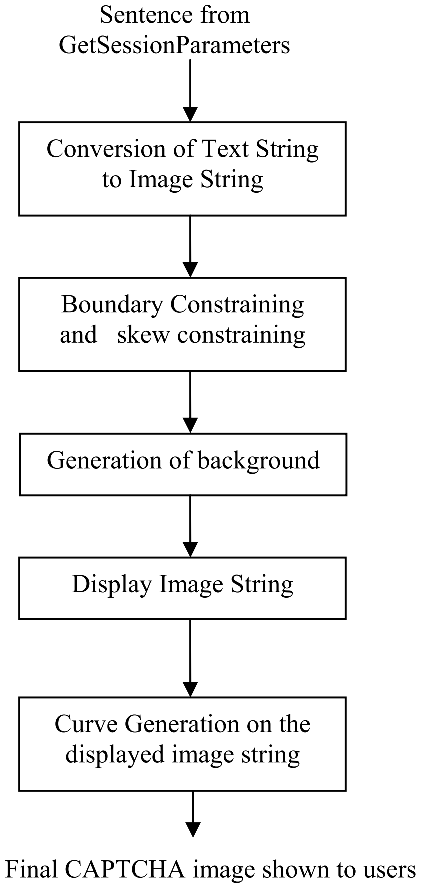

Sentence from
GetSessionParameters
Conversion of Text String to Image String
Boundary Constraining and skew constraining
Generation of background
Display Image String
Curve Generation on the displayed image string
Final CAPTCHA image shown to users

Fig. 4 Generation & Display of CAPTCHA text

- **Generation of Background:** A background of randomly colored regions and random sizes is generated. To increase the complexity in order to prevent an automated program to read the text, several gradients are introduced in the background of the image. Moreover, the rectangle in which the image is displayed is itself divided into three different regions. Each of the regions is filled with randomly chosen colors. However, to maintain consistency, the positioning and the intensity of the gradients is maintained same in all the regions.

- **Curve generation on the displayed image string:** A sinusoidal curve is drawn over the CAPTCHA image. This is done by choosing the points across the breadth of the bounding box. Since the bounding box is generated randomly and varies with each execution, the position of the curve also varies along with it, thereby making sure that the automated program cannot fix any pattern for matching.

❖ *Recording the input speech*

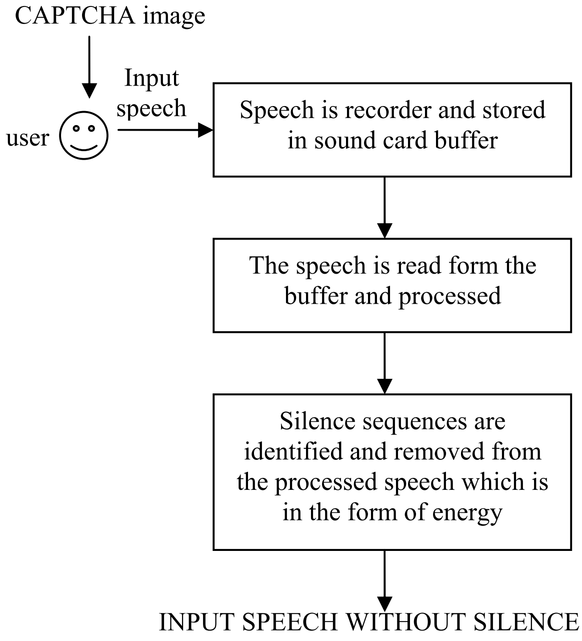
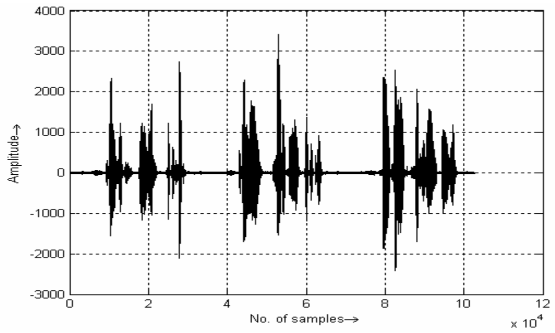

CAPTCHA image
user
Input speech
Speech is recorder and stored in sound card buffer
The speech is read form the buffer and processed
Silence sequences are identified and removed from the processed speech which is in the form of energy
INPUT SPEECH WITHOUT SILENCE

Fig. 5 Recording of input speech

- **Recording the continuous input speech of the user:** The user should read out the displayed CAPTCHA image as clearly as possible. The recorded speech is read from the sound card into a buffer. The recorded voice can be saved in different formats, each of which is handled in different ways. We employ the .wav format to store the recorded voices [5]. The structure of this format contains three parts – header, format and data [6]. We have three structures to associate the format with their related fields. Multimedia system header files are used to take care of the processing on the recorded speech.

- **Removal of silence sequences from the input speech:** The recorded speech is treated as energy to detect the silence occurrences [7]. The continuous energy is divided into several fixed-size frames and the average energy level for the frame is calculated. Using these average values of each frame, a smooth curve is plotted and it is differentiated to find out the threshold



point i.e. the energy levels below the threshold point are considered as silence. After the entire file has been broken down into frames and the silence sequences identified (both sample-wise and time-wise), the frames having average value above the threshold are written to a file. The background noise is assumed to be Gaussian in nature [8]. The speech signal too may have different types of noises in it. This file now is the recorded voice without silence sequences.

❖ *Recognition*

Recognition is done by using the HTK Toolkit.

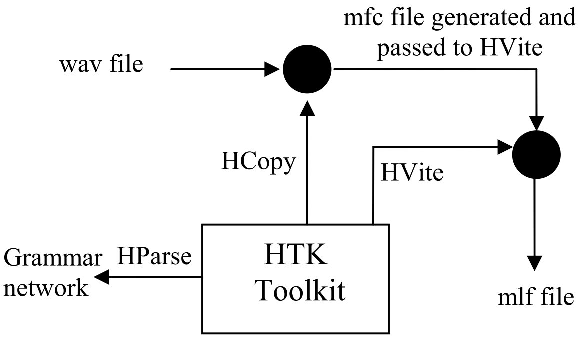


Fig. 6 Recognition

- **Speech Parameterization:** For the purpose of speech recognition, parameters need to be extracted from the input speech which is the quantized signal amplitudes values. The amplitude values are read from the wav file and are put into overlapping blocks. Each such block is processed to generate a single MFCC (Mel Frequency Cepstral Coefficients) vector [9]. The sequences of the MFCC vectors corresponding to all blocks are stored in the .mfc file. This process is called parameterization of the input speech and these vectors are used by the speech recognizer.

- **Grammar Network Generation:** The process of speech recognition tries to match the input data (MFCC vectors computed from wav files) of the model of the most likely matching word. This process is called decoding and the recognizer has to try out all possible paths corresponding to all probable words that are likely to be present in the input speech. The decoding can be enhanced by using Layered decoding network architecture [10]. The network of all possible paths taken by the decoder is the grammar network.

❖ *Verification*

- **Ordering of Keywords:** The keywords for a particular sentence are generated in the same order in which they occur in the sentence. The decoded speech which is in the form of a file (.mlf) is the input to this module. All the keywords generated for this session are stored in an array. The file is then scanned for these keywords in the same order in which they occur, hence maintaining the order of the keywords. At any mismatch, the verification process is stopped and the user is denied.

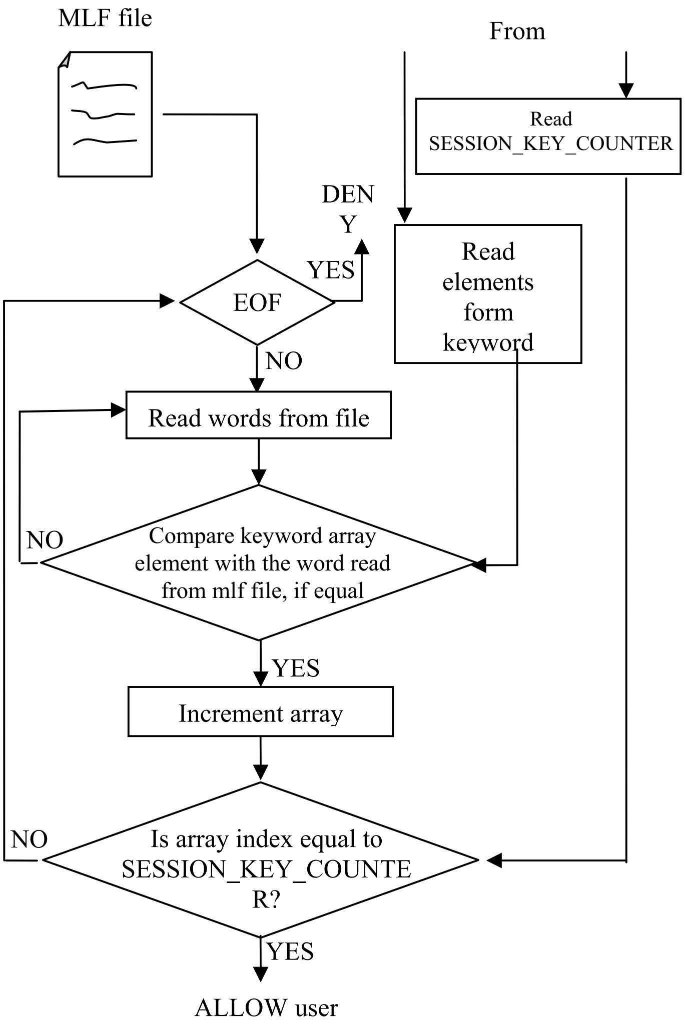

MLF file
From
Read SESSION_KEY_COUNTER
DEN Y
YES
EOF
Read elements form keyword
NO
Read words from file
NO
Compare keyword array element with the word read from mlf file, if equal
YES
Increment array
NO
Is array index equal to SESSION_KEY_COUNTE R?
YES
ALLOW user

Fig. 7 Verification

- **Keyword comparison:** Each keyword in the mlf file is compared with the corresponding keyword in the array. The keywords should occur in the same order in the mlf file and the array. For each match, the array index is incremented. At any mismatch, the verification process is stopped thus denying the user . This ensures that the user has to read out the sentence as displayed. He cannot mix up the words or change the order of utterance.

- **Allowing/Denying the user:** If the final array index value is not equal to the number of keywords selected for the session, the user is denied. Otherwise, he is allowed.

IV. CONCLUSION & FUTURE ENHANCEMENTS

The increasing number of attacks on online transaction systems by automatic programs or “bots” has proved to be an important security concern in recent times. The existing mechanisms to curb these attacks are in the form of CAPTCHA’s, but, they too have been vulnerable to attacks in the recent times. Hence, there arises a need for a stronger and more secure mechanism. Speech recognition is a good solution to this problem. When CAPTCHA technology and speech recognition work in combination, the system can determine the presence of a human as well as identify the individual.

Limitations

This work is limited to utterance verification i.e. verifying what the user has spoken. It does not take care of individual identification. The performance is dependent on the quality of training done.

Future enhancements

- The product can be enhanced to perform user authentication based on speech. The same speech given for utterance verification can be used for this purpose.
- Weighted scoring of keywords can be used for robust performance
- Presently, the system can be used only by people who know English. The system can be enhanced to accommodate more languages.

VI. ACKNOWLEDGMENT

Our work was the result of the encouragement of many people who helped in shaping it and provided feedback, direction valuable support. It is with hearty gratitude that we acknowledge their contributions to our work. We would like to thank our guide *Dr. R Vasantha*, Department of Information Science & Engineering, RVCE, for the constant help and support extended towards us during the course of the work.

REFERENCES

[1] Kumar Chellapilla, Kevin Larson, Patrice Simard, Mary Czerwinski (2005). "Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs)”

[2] Jeff Prosise (2001). “Programming Windows with MFC”

[3] Moni Naor (1996-09-13). "Verification of a human in the loop or Identification via the Turing Test”

[4] Rose, R.C.; Paul, D.B. “*A hidden Markov model based keyword recognition system”,* ICASSP-90

[5] http://www.fcla.edu/digitalArchive/pdfs/ action_plan_bgrounds/wav.pdf

[6] http://ccrma.stanford.edu/courses/422/projects/ Wave Format

[7] G. Saha, Sandipan Chakroborty, Suman Senapati, *“A New Silence Removal and Endpoint Detection Algorithm forSpeech and Speaker Recognition Applications”.* Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Khragpur

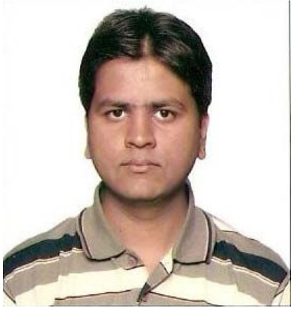
[8] L.R.Rabiner and R.W.Schafer*,”Digital Processing of Speech Signals”,* First Edition, Chapter 4, Pearson Education, Prentice-Hall.

[9] Christine Englund (2004), “Speech recognition in the JAS 39 aircraft-adaptation at different G-loads”, Master Thesis in Speech Technology

[10] Qiru Zhou and Wu chou, “An Approach to Continuous Speech Recognition Based on Layered Self-Adjusting Decoding Graph”, Bell Laboratories, Lucent Technologies.

Authors:

Ashwin S Kumar is with the Department of Information Science, R V College of Engineering, Bangalore. (Phone: 91-80-23217481; e-mail: ashwinskumar@gmail.com).



D B Mahesh Kumar is with the Department of Information Science, R V College of Engineering, Bangalore. (Phone: 91-09341616713; e-mail: rvce.mahesh@gmail.com).

Mayank Kumar is with the Department of Information Science, R V College of Engineering, Bangalore. (Phone: 91-09986155825; e-mail: mayankiaf@gmail.com).