

## VOCAL CODE

Gyan Prakash<sup>1</sup>, Aditya Bapat<sup>2</sup>, Ajay Shewale<sup>3</sup>, Sona R Pawara<sup>4</sup>

<sup>1,2,3</sup>Student, Department of Computer, Sinhgad Academy of Engineering, Pune, India

<sup>4</sup>Professor, Department of Computer, Sinhgad Academy of Engineering, Pune, India

\*\*\*

**Abstract** - Software development by means of programming languages involve keyboard for input. All programming languages are mostly text-oriented. This text familiarizes nature of programming languages is an obstacle to persons suffering from arms disability. Someone having dazzling mind and potential for programming skills, but suffering from arm injuries or being disabled could not become a programmer. To be a good developer it is important to remember the syntax and keywords of a programming language.

In our project, we propose a methodology for Java programming language where a programmer will speak and code in Java will be written accordingly.

**Key Words:** *VOCAL CODE, SPEECH CODE, REPETITIVE STRAIN INJURIES (RSI), HMM, SPEECH RECOGNITION*

### 1. INTRODUCTION

VocalCode supports Java programming language. The project implementation consists mainly conversion of speech to text using voice recognition algorithms and then that text for coding purposes for them who are suffering from repetitive strain injuries (RSI) and related disabilities that make typing difficult or impossible. It is done by adding commands (methods, classes, conditional constructs, loop templates, etc.) and their spoken formats to the Vocal code. And we are using HMM models for speech recognition and MFCC for feature extraction. Wherever possible, we have kept the spoken formats for Java language consistent with formats in Java language. We have added support for common commands (println, main method etc.) and set them up to do a lot of automatic typing for the user. Speech recognition software has helped many people who have difficulty in typing. Commercial speech recognition programs serve the purpose of dictation of text in a natural language, such as English. Programming languages were never supposed to be Vocal and have a lot of punctuation, unusual spelling and capitalization ("println", "compute Length"), and non-standard symbols (++ , --, && etc.). Due to this standard speech recognition software cannot be simply used to write programs. Java speech support has the latent to greatly help the programmers who have difficulty using their hands. Java is one of the most popular programming languages. Hence Introducing speech typing for Java Language will be beneficial for all programmers.

### 2. LITERATURE SURVEY

[1] Soon Suck Jarng :The HMM voice recognition algorithm is explained and the importance of voice information DB is revealed for better improvement of voice recognition rate.

[2] Stephen C. Arnold: Vocal Programming. A system that enables a person to program without typing is needed because of the high incidents of repetitive stress injuries among people who program. This paper presents a design for a system that generates environments that enables people to program by voice and a method of determining if the system is successful. It also shows how this generator can be used to support entering data and writing XML documents.

[3] Lawrence R. Rabiner :A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In this paper they attempt to carefully and mathematically review the theoretical aspects of this type of statistical modeling and show how they have been applied to selected problems in machine recognition of speech.

[4] Hyan-Soo Bae, Ho-Jin Lee, Suk-Gyu Lee :Voice Recognition based on adaptive MFCC and deep learning. In this paper, they propose an enhanced voice recognition method using Adaptive MFCC and Deep Learning. To improve the voice recognition rate, it is important to extract the audio data from original signal. However the existing Algorithms which is used to remove the noise of particular band deteriorate the audio signal.

[5] Yen-Huann Goh, Paramesran Raveendran: HMM-based speech recognition using adaptive framing. A common approach in mapping a signal to discrete events is to define a set of symbols that correspond to useful acoustic features of the signal over a short constant time interval. This paper proposes a hidden Markov models (HMM) based speech recognition by using cepstrum feature of the signal over adaptive time interval.

[6] Mark Gales and Steve Young: The Application of Hidden Markov Models in Speech Recognition. Hidden Markov Models (HMMs) provide a simple and effective framework for modelling time-varying spectral vector sequences. As a consequence, almost all present day large vocabulary continuous speech recognition (LVCSR) systems are based on HMMs.

### 3. PROPOSED SYSTEM

Programming surroundings can create annoying barriers for the rising numbers of software developers that suffer from repetitive strain injuries (RSI) and related disabilities that make the typing difficult or impossible. Not only is the software development process contain somewhat text rigorous activities like program composition, editing, and navigation, but the tools used for programming are also operated textually.

The main purpose of the project is to reduce the efforts to write long codes and reduce the physical problems associated with it and also to make it possible for more and more people to write codes.

Our system is mainly divided into three Modules as shown in Figure 1.

- GUI(Graphical User Interface)
- Speech To Text Convertor
- Code Generator

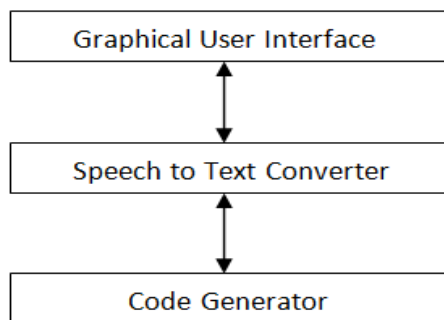


Figure 1: Modules for Vocal coding

Let's have a brief overview of the high level working of our system, before getting into the details of the system. The GUI module provides an interface for users to interact with the system. Speech to text converter is responsible for converting each listened word to text. A user will speak code using microphone and text converter will convert this spoken code to text. For this converted text Java syntax and semantics are applied to this text to generate standardized Java code.

#### 3.1 Graphical User Interface

GUI gives the visual appearance of the virtual file system to the end user. GUI consist of color schemes, layout, working and behavior provides easy access to common operations and gives appealing look. Standard Toolbars, popup menus and shortcut keys make operation of software easy for all type of users. Easy to Use, Easy accessibility to functions and Appealing appearance are the main features of GUI.

#### 3.2 Speech to Text Converter

The core part of our system is speech to text conversion. For speech recognition, first we are developing a speech recognition engine software program, and then record a person's or several people's voice. Subsequently we should divide the speech into sub-units (syllables), and then the syllables should go through the learning process. Increasing the number of samples of the same syllable during learning tends to increase the recognition rate. At the end of the learning process, the re-recorded syllables go through the speech recognition process. For this system we are implementing HMM method, and simple syllables were used for the recognition process. In order to achieve a high recognition rate for different syllables, significant quantitative information of syllables is required. And for that information we require feature extraction method MFCC(Mel Frequency Cepstral Coefficients).

The speech recognition system may be viewed as working in a four stages as shown in Figure 2.

- i. Feature extraction
- ii. Pattern training
- iii. Pattern Matching.
- iv. Decision logic

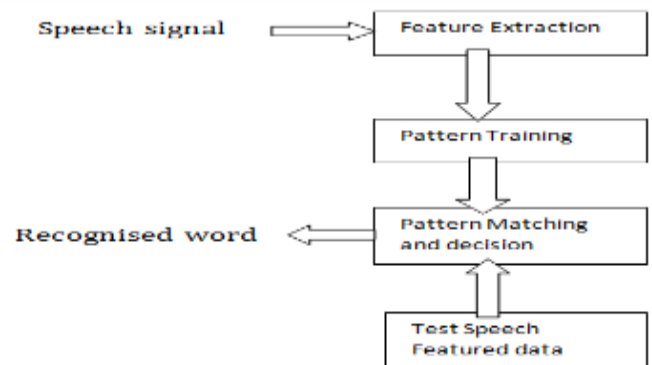


Figure 2: Speech Recognition System

The feature extraction process is implemented using Mel Frequency Cepstral Coefficients (MFCC) in which speech features are extracted for all the speech samples. Then all these features are given to pattern trainer for training and are trained by HMM to create HMM model for each word. Then viterbi decoding will be used to select the one with maximum likelihood which is nothing but recognized word.

##### 3.2.1 HMM (Hidden Markov Model)

HMM (Hidden Markov Model) recognition algorithm, for every new input voice signal, voice feature parameters are generated which are used in the learning process to create

a new HMM model. So with each new HMM model created for every word, during the testing phase, all these models are compared with the test word to find out the matching voice sample.

In HMM model for every new voice that is added to the model, a new individual HMM model needs to be created, and each model should be compared with all the existing HMM models to get a match. HMM method is fast in initial training, and when a new voice information is added into the HMM database, only the new voice is used in the training process to create a new HMM model. Compared to the neural network algorithm, for a large number of speech samples, the HMM algorithm provides a higher speech recognition rate.

In this recognition algorithm, in order to increase the recognition rate, unique feature parameters (Feature) of the signal should be extracted. Even the same voices with the same meaning differ in sound intensity, pitch, and timbre. The voice waveform varies in vocalizing speed, personnel style, and is masked by environmental noise. Speech itself strongly depends on the language. Rate of speech and ambient noise, are considered to be the biggest factors that reduce recognition rate and ways to overcome these challenges are presented.

### 3.2.2 FEATURE PARAMETER EXTRACTION

First, an important issue in speech recognition algorithm is to identify the unique features of speech and to extract the quantitative parameters, which are interwoven in the selection of parameters. Until now, MFCC (Mel-Frequency Cepstral Coefficients) parameters are the most commonly used, but extraction of new parameters of the speech signal can dramatically improve voice recognition. MFCC calculated from a given speech signal to know the hourly cepstrum is usually expressed as the coefficient matrix. MFCC feature parameters are extracted from the voice signal.

### 3.3 Code Generator

Code generator is the module that actually generates Java code from listened words. As a first step, we find a list of reserved words of Java. Now for each reserved word we find words with similar sound. For example "while" have following words with similar sound. "lloyd", "while", "white", "wine" and "voice ". After this, for each reserved word we developed a separate list structure of words with similar sound in Java. When a user speaks a reserved word that word will be converted to text and this text will be matched to the elements of list structure. If a match is found converted text is replaced with that reserved word. If no match is found text is written as it is. Now if this text is wrong and user wants to remove that word user will speak "incorrect". A list structure is also maintained for same utterances of "incorrect". If spoken word is matched with

that same utterance then that word is removed. At the same time if a match is found and that reserved word has special program construct then that program construct is also generated simultaneously.

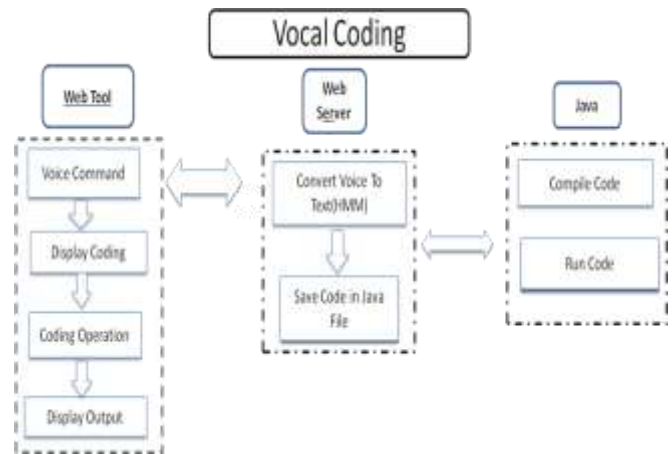


Fig -1: Architecture

### 3. CONCLUSIONS AND FUTURE SCOPE

This paper provides possible approaches that are best suitable to develop a speech recognition system to learn programming concepts for people suffering from repetitive strain injuries (RSI) and related disabilities that make typing difficult or impossible . Over the last few decades, advancements in web phone technologies have reached an impressive level. With the advancement of web technology, the society has changed drastically.

This research provides a solution to these problems by developing the system "Vocal Code" to take them forward. This system mainly focused on developing software that acts as an assistant, especially for RSI and related disabilities people. "Vocal Code" is an intelligent system, based on speech recognition and user interaction system. Capturing and recognizing the given sound input is one of the most important features. Moreover, also providing the ability to use anywhere anytime without hard efforts. The ultimate goal of this research is to make provision for people who are unable to type code, solve their programming learning related issues for empowering the differently able. The application that building may contain a limited number of datasets with the sounds and words and also hope to get accuracy more than 90%. As this system developed for RSI affected and handicapped people as a web application, this will be a big help to them. "Vocal Code" would be a baseline for future young inventors to use for their researchers. By using this framework as a building block, they can continue building what they want on top of this. This will be an open door for future developers as a platform for voice recognition.

## 5. REQUIREMENT SPECIFICATIONS

**Hardware Requirements:** Pycharm IDE or Visual studio runs on operating systems that support the Java VM (Virtual Machine) and has been tested on the platforms listed below:

**Processor:** 2.40GHz Intel i3, i5 **equivalenMemory:** 2GB

### Software Requirements:

Java Development Kit (JDK) which consists of the Java Runtime Environment and developer tools for compiling, Software used are Pycharm or Visual Studio.

## ACKNOWLEDGEMENT

We take this opportunity to thank all the people involved in the making of this paper. We want to especially thank our respected guide, Prof. Sona R Pawara for her guidance and encouragement, which has assisted us to achieve our goal. Her valuable advice has helped us throughout. Our Head of Department Prof. B.B Gite and Project Coordinator Mr. Santosh Shelke has also been very helpful and we are grateful for the support he provided us with. Last but not the least we would like to convey our gratitude to all the teaching and non-teaching staff members of our department, our friends and families for their valuable suggestions and support.

## REFERENCES

- [1] Soon Suck Jarng: The HMM voice recognition algorithm. DOC: 26-29 April 2011, IEEE, ISSN: 2162-9048
- [2] Cristian IONIȚA "Building Domain Specific Languages for Voice Recognition Applications", Revista Informatica Economica nr. 2(46)/2008.
- [3] Lawrence R. Rabiner, IEEE 'A Tutorial On Hidden Markov Model And Selected Applications In Speech Recognition, Proceedings Of The IEEE, Vol. 77, No. 2, February 1989.
- [4] Hyan-Soo Bae, Ho-Jin Lee, Suk-Gyu Lee, Voice Recogniton based on adaptive MFCC and deep learning. 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)
- [5] Yen-Huann Goh, Paramesran Raveendran, HMM-based speech recognition using adaptive framing, IEEE ISBN: 978-1-4244-4546-2. Date of Conference: 23-26 Jan. 2009
- [6] Foundations and Trends<sup>®</sup>R in Signal Processing Vol. 1, No. 3 (2007) 195–304 <sup>©</sup>c 2008 M. Gales and S. Young DOI: 10.1561/2000000004