Speaker Independent Recognition System with Mouse Movements

Dr.R.L.K.Venkateswarlu[#1], Dr. R. Vasantha Kumari[*2], G.Vani JayaSri[#3]

[1]*Dean of Research,* [3]*Associate Professor*
[#] *Sasi Institute of Technology & Engineering, Tadepalligudem, India*
[2]*Principal *Perunthalaivar Kamarajar Arts College, Puducherry 605 107*

Abstract - Speech recognition is potentially a multi-billion dollar industry in the near future. It is a natural alternative interface to computers for people with limited mobility in their arms and hands, sight, hearing limitation. For most current voice-mail systems, one has to follow series of touch-tone button presses to navigate through a hierarchical menu. Speech Recognition has the potential to cut through the menu hierarchy. Recently, neural networks have been considered for speech recognition tasks since in many cases they have shown comparable performance than the traditional approaches. There are two in-built threads in the recognition system. Thread 1 maintains the details about input acquisition where as thread 2 contains the classifier and decoder. The classifier used in this research is Radial Basis Function Neural Networks. The HMM graph is used as a decoder. The objective of the research is to make sure that the system is free from bugs. 100% accuracy is achieved by the recognition system.

Keyword - Thread, Recognizer, Hidden Markov Model, Radial Basis Function, Mouse Movements

1. I. INTRODUCTION

A. Voice-controlled human-computer interface

A voice-controlled human-computer interface has been designed that enables severely handicapped individuals to operate a computer. This system consists of six main blocks: the keyboard and the mouse layouts; the D6106 speech recognizer; the headset incorporated with a microphone and two mercury switches; the keyboard circuit and the mouse circuits; the mouse control circuit; and the microcontroller. A prototype system has been built and tested.

B. User can fully manipulate the computer via voice commands

The user interface, in the industrial design field of human–machine interaction, is the space where interaction between humans and machines occurs. The goal of interaction between a human and a machine at the user interface is effective operation and control of the machine, and feedback from the machine which aids the operator in making operational decisions.

C. Ideal method for disabled persons who cannot use conventional computer interfaces such as keyboard, mouse.

In human-computer interaction, computer accessibility (also known as Accessible computing) refers to the accessibility of a computer system to all people, regardless of disability or severity of impairment. It is largely a software concern; when software, hardware, or a combination of hardware and software, is used to enable use of a computer by a person with a disability or impairment, this is known as Assistive Technology.

2. II. SPEECH RECOGNITION

Speech recognition, often called automatic speech recognition, is the process by which a computer recognizes what a person said. If you are familiar with speech recognition, it 's probably from applications based around the telephone. If you've ever called a company and a computer asked you to say the name of the person you want to talk to, the computer recognized the name you said through speech recognition.

This is very different from a computer actually understanding what you said. When two people speak to one another, they both recognize the words and the meaning behind them. Computers, on the other hand, are only capable of the first thing: they can recognize individual words and phrases, but they don't really understand speech in the same way humans do.

Speech recognition is still useful, however, because we don't need computers to actually carry on conversations with us. we just need to give them commands. When you type a word or phrase, the computer doesn't actually understand English, but it recognizes the command and software tells the computer what to do when that command is recognized.

The same is true of speech recognition software. Users speak commands that are recognized by a piece of software called the speech engine. The speech engine then tells the speech application what the user said, and the application determines what to do next.

In speech applications such as dictation software, the application's response to hearing a recognized word may be to write it in a word processor. In an interactive voice response system, the speech application might recognize a person's name and route a caller to that person's phone.

Speech recognition is also different from voice recognition, though many people use the terms interchangeably. In a technical sense, voice recognition is strictly about trying to recognize individual voices, not what the speaker said. It is a form of biometrics, the process of identifying a specific individual, often used for security applications.

Because we all have distinct speaking styles, this is why you can tell your mom's voice from your favorite radio talk show host's, computers can take a sample of speech and analyze it for distinct characteristics, creating a "voice print" that is unique [5]. A common voice recognition system might make the user speak a password. It would then compare the speaker's voice print to a stored voice print and authenticate the user if they matched.

Though speech recognition uses some of the same fundamental technology as voice recognition, it is different because it does not try to identify individuals. Rather it tries to recognize what individuals say. It's the difference between knowing who is speaking and what is said. Though they vary greatly, speech engines generally use a similar process to figure out what a speaker said:

1. The engine loads a list of letters to be recognized. This list of letters is called a grammar.
2. The engine loads audio from the speaker. This audio is represented as a waveform, essentially the mathematical representation of sound.
3. The engine compares the waveform to its own acoustic models. These are databases that contain information about the waveforms of individual sounds and are what allow the engine to recognize speech.
4. The engine compares the letters in the grammar to the results it obtained from searching its acoustic models.
5. It then determines which letters in the grammar the audio most closely matches and returns a result.

A. Applications

The applications of speech recognition system are widely found in Dictation, command and Control, Medical and Embedded Systems.

III. SPEAKER INDEPENDENT

In Speaker Independent mode the training data is not same as tested data.

IV. NEURAL NETWORKS

Neural networks have been given serious consideration for speech recognition problems due to the following reasons

1) Neural network can readily a massive degree of parallel computation. Because a neural network is a parallel structure of simple, identical, computational elements, it should be clear that they are prime candidates for massively parallel (analog or digital) computation.
2) Neural networks possess a great deal of robustness or fault tolerance. Since the "information" embedded in the neural is "spread" to every computational element within the network, the structure is inherently among the least sensitive of networks to noise or defects within the structure.
3) The connection weights of the network need not be constrained to be fixed, they can be adapted in real time to improve performance. This is the basis of the concept of adaptive learning, which is inherent in the neural network structure.
4) Because of the nonlinearity with each computational element, a sufficiently large neural network can approximate (arbitrarily closely) any nonlineariry or nonlinear dynamical system. Hence neural networks provide a convenient way of implementing nonlinear transformations between arbitrary inputs and outputs and are often more efficient than alternative physical implementations of the nonlinearity.

V. HIDDEN MARKOV MODEL

HMM represents speech by a sequence of states, each representing a piece of the input signal. The states of the HMM correspond to phones, biphones or triphones. At each state, there is a probability distribution for each of the possible letters, and a transition probability to the next state. The speech recognition processes then boils down to finding the most probable path. The training procedure for the HMM-based recognizer is more complex than the DTW-based recognizer (Rabiner et al., 1989; Lee, 1988; Woodland et al., 1994; Huang, 1992).

A. The advantages of an HMM-based approach are

- It is easy to incorporate other information, such as speech and language models.

• Continuous HMM is powerful for continuous speech recognition.

B. *The disadvantages of HMM-based approach are*

• The HMM probability density models (discrete, continuous, and semi-continuous) have suboptimal modeling accuracy. Specifically, discrete density HMMs suffer from quantization errors, while continuous or semi-continuous density HMMs suffer from model mismatch.

• The Maximum Likelihood training criterion leads to poor discrimination between the acoustic models. Discrimination can be improved using the Maximum Mutual Information training criterion, but this is more complex and difficult to implement properly.

VI. Radial Basis Function Neural Network

The core of a speech recognition system is the recognition engine. The one chosen in the paper is the Radial Basis Function Neural Network (RBF). This is a static two neuron layers feed forward network with the first layer L1, called the hidden layer and the second layer, L2, called the output layer. L1 consists of kernel nodes that compute a localized and radially symmetric basis functions.[1]

The pattern recognition approach avoids explicit segmentation and labeling of speech. Instead, the recognizer used the patterns directly. It is based on comparing a given speech pattern with previously stored ones. The way speech patterns are formulated in the reference database affects the performance of the recognizer. In general, there are two common representations. The output y of an input vector x to a (RBF) neural network with H nodes in the hidden layer is governed by:

$$y = \sum_{h=0}^{H-1} w_h \, \emptyset_h$$

Where w_h are linear weights $\emptyset_h$ are the radial symmetric basis functions. Each one of these functions is characterized by its center c_h and by its spread or width σ_h. The range of each of these functions is [0, 1].

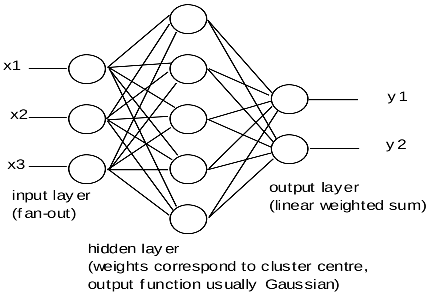

Fig 1: Radial Basis Function Neural Network

A. *Architecture*

Once the input vector x is presented to the network, each neuron in the layer L1 will output a values according to how close the input vector is to its weight vector. The more similar the input is to the neuron's weight vector, the closer to 1 is the neuron's output and vice versa. If a neuron has an output 1, then its output weights in the second layer L2 pass their values to the neurons of L2. The similarity between the input and the weights is usually measured by a basis function in the hidden nodes.

VII. Hidden Markov Model

Definition of Hidden Markov Model:

The Hidden Markov Model is a finite set of states, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state an outcome or observation can be generated, according to the associated probability distribution. It is only the outcome, not the state visible to an external observer and therefore states are ``hidden" to the outside; hence the name Hidden Markov Model.

In order to define an HMM completely, following elements are needed.

- The number of states of the model, N.
- The number of observation symbols in the alphabet, M. If the observations are continuous then M is infinite.
- A set of state transition probabilities $\Lambda = \{a_{ij}\}$

$$a_{ij} = p\{q_{t+1} = j \mid q_t = i\}, 1 \le i, j \le N,$$

where q_t denotes the current state. Transition probabilities should satisfy the normal stochastic constraints, $a_{ij} \ge 0, 1 \le i, j \le$ and

$$\sum_{j=1}^{N} a_{ij} = 1, 1 \le i \le N$$

A probability distribution in each of the states,

$$B = \{b_j(k)\} .$$

$$b_j(k) = p\{O_t = V_k \mid q_t = j\}, \quad 1 \le j \le N, \quad 1 \le k \le M$$

where V_k denotes the k^{th} observation symbol in the alphabet, and O_t the current parameter vector. Following stochastic constraints must be satisfied.

$b_j(k) = 0, 1 \leq j \leq N, \quad 1 \leq k \leq M$ and

$$\sum_{k=1}^{M} b_j(k) = 1, \qquad 1 \leq j \leq N$$

If the observations are continuous then we will have to use a continuous probability density function, instead of a set of discrete probabilities. In this case we specify the parameters of the probability density function. Usually the probability density is approximated by a weighted sum of M Gaussian distributions $\mathcal{N}$,

$$b_j(O_t) = \sum_{m=1}^{M} c_{jm} N(\mu_{jm}, \Sigma_{jm}, O_t)$$

where,

C_{jm} = weighting coefficients

μ_{jm} = mean vectors

Σ_{jm} = Covariance matrices

C_{jm} = should satisfy the stochastic constrains,

$C_{jm} \geq 0, 1 \leq j \leq N, \; 1 \leq m \leq M$ and

$$\sum_{m=1}^{M} C_{jm} = 1, \qquad 1 \leq j \leq N$$

The initial state distribution, $\pi = \{\pi_i\}$. where, $\pi_i = p\{q_1 = i\}, \; 1 \leq i \leq N$.

Therefore we can use the compact notation $\lambda = (\Lambda, B, \pi)$

to denote an HMM with discrete probability distributions, while

$\lambda = (\Lambda, C_{jm}, \mu_{jm}, \Sigma_{jm}, \pi)$ to denote one with continuous densities.

As mentioned earlier, ASR problem can be attacked from two sides; namely

1. From the side of speech generation
2. From the side of speech perception

The Hidden Markov Model (HMM) is a result of the attempt to model the speech generation statistically, and thus belongs to the first category above. During the past several years it has become the most successful speech model used in ASR. The main reason for this success is it's wonderful ability to characterize the speech signal in a mathematically tractable way.

In a typical HMM based ASR system, the HMM stage is proceeded by the preprocessing (parameter extraction) stages.[9]The parameter vectors can be supplied to the HMM, either in vector quantized form or in raw continuous form. It can be designed HMMs to handle any of the cases, but important point is how the HMM deals with the stochastic nature of the amplitudes of the feature vectors which is superimposed on the time stochasticity.

Evaluation problem can be used for isolated (word) recognition. Decoding problem is related to the continuous recognition as well as to the segmentation. Learning problem must be solved, if we want to train an HMM for the subsequent use of recognition tasks.

A. Use of HMMs in continuous recognition

In the isolated mode we used one HMM for each of the speech unit. But in the continuous case this is not possible because a sequence of connected speech units, which is usually called a sentence, is to be recognized and hence the number of possible sentences may be prohibitively large even for a small vocabulary. In addition to this, there are two other fundamental problems associated with continuous recognition.

(1) We do not know the end points of the speech units contained in the sentence.

(2) We do not know how many speech units are contained in the sentence.

VIII. Existing System

In the present scenario the data/information is being retrieve by manual process that means it uses the mp3 files or any audio formatted supporting files ,but in this kind of a process the data /information static data so the users want to access the d/information in a dynamic process is not possible so we need to overwrite the audio files / information with the new data due to this it is a tedious task to the users to implements this but ,we use the centralized location physical storage system so that the users can access the data. The existing system is implemented by Gaussian mixture model.

IX. Proposed System

To overcome the limitations and defects are the previous existing system we considered Hybrid approach as the best suited system as the architecture design is concerned here, i.e. the user is going to do the mouse movement operations with a voice in the second feature using the synthesizer feature the speech recognition technique will be applied through the mike device so the system will receive the commands given by the user using the NN technique.[7]

Power spectral density function (PSD) shows the strength of the variations (energy) as a function of frequency. In other words, it shows at which frequencies variations are strong and at which frequencies variations are weak. The unit of PSD is energy per frequency (width) and you can obtain energy within a specific frequency range by integrating PSD within that frequency range. Computation of PSD is done directly by the method called FFT or computing autocorrelation function and then transforming So using the psd accent, dictionary, and grammar info can betaken into consideration.

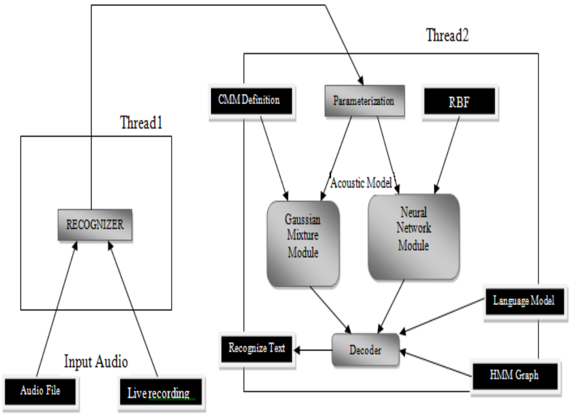

Fig 2: Proposed System Architecture

A. Architecture Description

This architecture contains two threads i.e; thread1 and thread2. Thread1 contains recognizer. The input of the recognizer is audio file or any live recording. The files can be save as .wav format. And than the recognizer output is the input of parameterization. Thread2 contains two parametrization methods

i) Gaussian Mixture Module
ii) Neural Network Module

In this we use neural network module. The classifier of this module RBF. The output of the classifiers is the input of the decoder. Decoder decodes this input by using HMM model. And then finally we get the recognized mouse movements like left, right, top, bottom, selecting and releasing.

B. System Design

Computer software design changes continuously as new methods; better analysis and broader understanding evolved. Software Design is at relatively early stage in its revolution. Therefore, Software Design methodology lacks the depth, flexibility and quantitative nature that are normally associated with more classical engineering disciplines. However techniques for software designs do exist, criteria for design qualities are available and design notation can be applied.

C. Execution Steps

1. Go to Start Menu -> Command Prompt ->
2. After Selecting the Command Prompt change the directory to C:\
3. After entering the C directory we change the directory to Speech Recog folder(C:\ cd Speech Recog)
4. And than again goto bin folder(C:\ cd bin)
5. Next step is compilation of the source code in java language(javac MouseMove.java)
6. Next step execution (java Mouse Move)
7. Finally we get the results.

X. TESTING

We have different types of testing. There are black-box testing, white-box testing, unit testing, integration testing, validation testing, system testing and acceptance testing. In this we perform unit testing.

A. Unit Testing

The unit-testing focuses on the smallest executable program units. Once these small units have been tested individually, they can be combined for performing integration testing. The modules were fragmented into several small units in this phase and tested individually for better error detection. Unit testing has been performed by the developers themselves.

B. Test cases

I tested the Cursor Movements in Speaker Independent Mode. The table shows the test outline and its results are

TABLE I
TEST LAYOUT

Test data	.wav	.wav	.wav
Test type	Algorithm Test		Interface Test
Test item	- Six letters	- Correctness	- User Interface
Expected output	In speaker independent mode for the letters a, i, o, u, k and m we expect the cursor movements like	Cursor movements are done in speaker independent mode, similar to what we expect.	No errors while a user runs this system. Great satisfaction with our system

	left, right, top, bottom, selecting and releasing.		
Test process	Running the system	Running the system and check the cursor movements with plain text results	Asking questions users periodically and using think aloud method about usability.

XI. MAINTENANCE

The objectives of this maintenance work are to make sure that the system gets into work all time without any bug. Provision must be for environmental changes which may affect the computer or software system. This is called the maintenance of the system. Nowadays there is the rapid change in the software world. Due to this rapid change, the system should be capable of adapting these changes. In our project the process can be added without affecting other parts of the system.

Maintenance plays a vital role. The system liable to accept any modification after its implementation. This system has been designed to favour all new changes. Doing this will not affect the system's performance or its accuracy. In the project system testing is made as follows: The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated. Then the web form level testing is made.

This is the final step in system life cycle. Here we implement the tested error-free system into real-life environment and make necessary changes, which runs in an online fashion. Here system maintenance is done every months or year based on company policies, and is checked for errors like runtime errors, long run errors and other maintenances like table verification and reports.

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Implementation is the process of converting a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization.

XII. OUTPUT SCREENS

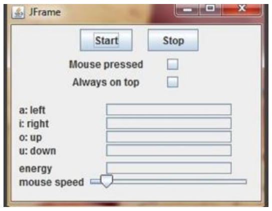

Fig 8a: Main Screen

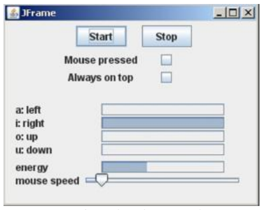

Fig 8b: Input Data for the letter 'i ' Output Screen

XIII. CONCLUSION

"Speech Recognition With Mouse Movements" was successfully designed and is tested for accuracy and quality. During this project, all the objectives have been accomplished and this project meets the needs of the Data Analyst. Users can choose various different dataset and detect the same cursor movements. The interface of the software is user friendly. Voice input which will save time and make life easier. It is capable of meeting the requirements of people unable to use their hands.

FUTURE WORK

This project use a .wav file format. File format and data type used in the code to detect cursor movements are fixed in this project but various formats and types might be allowable. In this we can develop only left click operations further we can perform right click operations also.

Besides improving the FE block and devising a more robust recognizer, the scope of the problem should be broadened to larger vocabularies, continuous speech, and more speakers. From this perspective, the results presented in this thesis are only preliminary. To develop our voice command in native languages besides English. To remove the acoustic signals that is not expected in our software.

REFERENCES

[1] Rabiner, L. and Juang, B. -H., 1993: Fundamentals of Speech Recognition, PTR Prentice Hall, San Francisco, NJ. pp.no 507.

[2] Gurney, K., 1997: An Introduction to Neural Networks, UCL Press, University of Sheffield, pp.no 234.

[3] Berthold, M.R., 1994: A Time Delay Radial Basis Function for Phoneme Recognition. Proc. Int. Conf. on Neural Network, Orlando, USA.

[4] Kandil N, Sood V K, Khorasani K and Patel R V, 1992: Fault identification in an AC–DC transmission system using neural networks, IEEE Transaction on Power System, 7(2):812–9.

[5] Park D C, El-Sharakawi M A and Ri Marks II, 1991: Electric load forecasting using artificial neural networks, IEEE Trans Power System, 6(2), pp 442–449.

[6] Picton, P. 2000: Neural Networks, Palgrave, NY .pp.no 195.(paper)

[7] Morgan, D. and Scolfield, C., 1991: Neural Networks and Speech Processing, Kluwer Academic Publishers, pp.no.391.

[8] Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. ASSP Magazine, IEEE, Vol. 3, No. 1, 4-16, ISSN: 0740-7467