

# Indian Institute of Technology, Guwahati



Department of Computer Science and Engineering

Project report

On

**“Speech based Contact Search App”**

Based on

Speech recognition system using HMM

Guided By :

Prof. P. K. Das

Submitted by:

Atul Bunkar - 214101011

Allam Eswar - 214101004

## **TABLE CONTENT**

- ◇ Abstract
- ◇ Introduction
- ◇ Methodology
- ◇ Flow of Application
- ◇ Working of Program
- ◇ Result

# **Abstract**

The application listens to the user for name of a person and displays the contact details of the spoken person's name which includes his/her phone number and email address.

# Introduction

This report focuses on interactive speech recognition programs between humans and computers.

The idea is to use the sounds of phonetics of speech to distinguish between the names, for example I have trained on names which can be easily differentiated by the model, like the name “Ishita” is recognized by the “shhh...” part of the name, “Mukund” is detected by the “moo” and “Koo” sound which differentiate these names easily from other names like “Aaru” which itself is distinguishable as a whole, names “Kirti” and “Virat” uses the vowel sound “e” at start but differs at the end. The names : Priyansh , Parul , Pranjal were difficult to separate while testing , and I must find a specific way to pronounce each one to get them correctly predicted. Thus, in the end it comes to how much of your sound and of the user in general has the model been trained on.

This project is just a minute start in the speech recognition world, to get people started on voice recognition.

## PROPOSED METHODOLOGY

Basic requirements to develop this project are as follows:

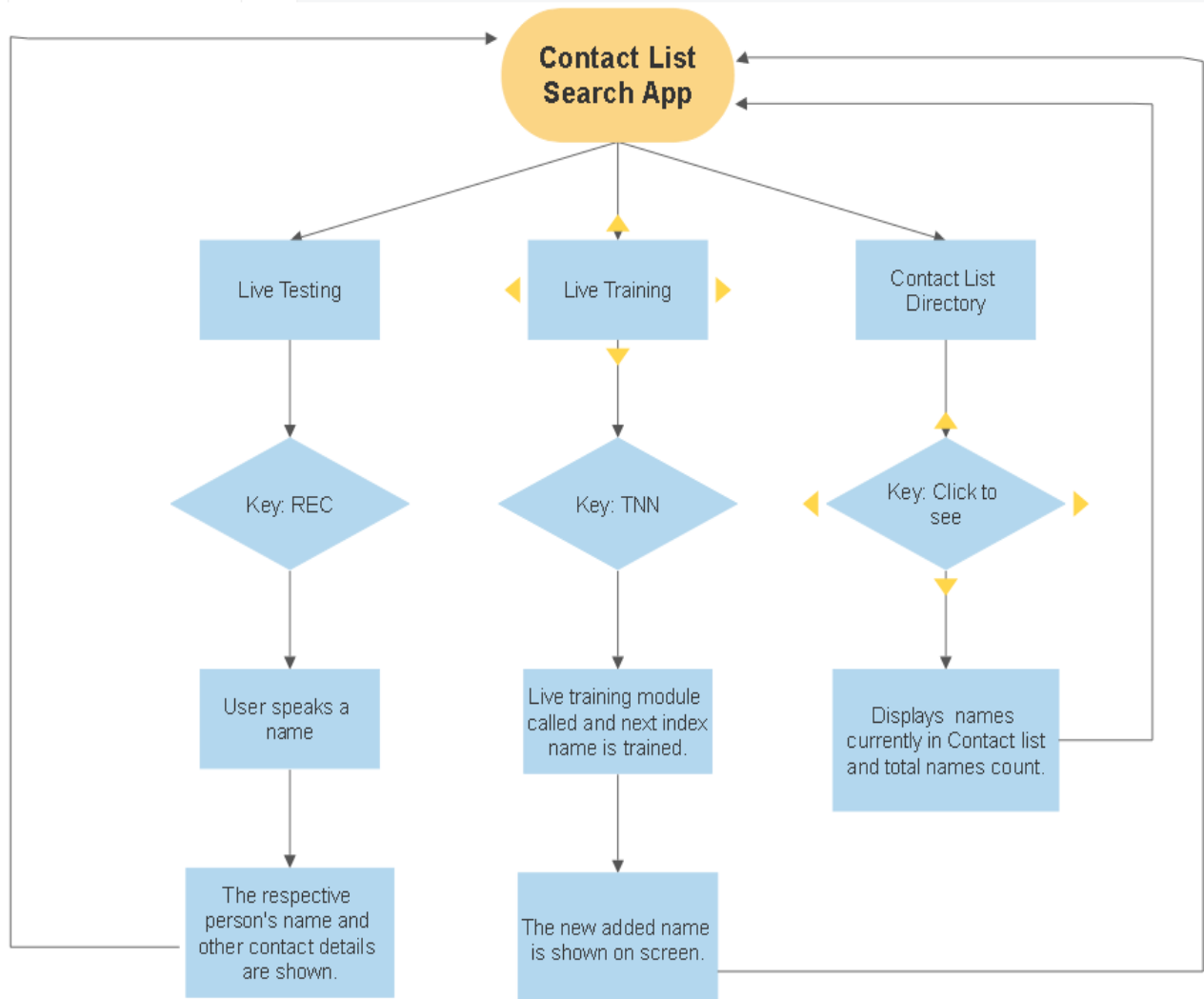
- ✓ Windows OS
- ✓ Microsoft Visual Studio 2010
- ✓ C++ 11 integrated with VS2010
- ✓ Recording Module

With the availability of above software, we further proceed in modelling the logic. The prerequisites of this project are

- ✓ Basic i/o operations on file
- ✓ Pre-processing of speech data
- ✓ Feature extraction
- ✓ Modelling of extracted feature
- ✓ Enhancing model

Above discussed topics are broadly elaborated in experimental setup section.

## Flow Diagram of Application



## Working:

The train, test data and models I have trained are stores based on the index of names given below (index, name) table:

0	1	2	3	4	5	6	7	8	9
Pranjal	Aaru	Ishita	Mukund	Virat	Sarthak	Madhuri	Priyansh	Rohan	Kirti

The names indexed 0 to 7 are already trained and their models stored in folder “model”. Names indexed 8 and 9 are live trained from application and shown after training. Their samples are stored in folder “live\_train”.

## Modules in Application:

### -> Live Testing:

The live recording.exe called which captures the user speech, preprocesses it, extracts observation sequence from codebook using nearest neighbour and checks with all trained models and returns index of model with highest  $P(\text{Observation}/\text{model})$ . This will be the model of uttered name.

## **-> Live Training:**

The training is called for remaining names (index 8,9) and they are trained. (After clicking TNN, wait for 10-15 secs), then the new trained name is shown. You can now live test for this name.

All words are trained with 25 utterance each, so for your own files (if training for own data), make sure to provide 25 utterances of each word and the file names has to be same as used here.

Working of Training( of all names ) :

- > Preprocess all names and extract Cepstral coefficients to generate a large Universe of Ci's.
- > From the Ci Universe, call the LBG method to generate a codebook (Vector Quantizer) of size 32.
- > Now pass each name utterance to this codebook and get observation sequence using NN for each utterance.
- > Create a base model (Feed Forward) to start the HMM.
- > The model is bad(biased) , Now we train/teach it by passing data to recognize and differentiate it.
- > The model is trained till it can't converge more and stored with respective name index.



## **RESULT**

We are getting contact details of 10 people which are already stored in a data file. Fetching of data based on user's speech is successfully implemented.