INDIAN INSTITUTE OF TECHNOLOGY, GUWAHATI

Department of Computer Science and Engineering

Project Report on

SPEECH BASED ONLINE PRODUCT SEARCH

Based on Speech Recognition System

Submitted to:

Prof. P. K. Das

Submitted by:

Paras Kangra (224101039)

Utkarsh Gaikwad(224101053)

Tejas Chandra Kareddula (224101052)

Acknowledgement

This project is being submitted as a requirement for course fulfillment of CS566 - Speech Processing. It is a pleasure to acknowledge our sense of gratitude to Prof. P.K. Das who guided us throughout the project work. Thank You For Your Guidance and Support For The Entire Course. We thank the Teaching Assistants who were always helpful in clearing doubts.

Yours sincerely:

1. Paras Kangra (224101039)
2. Utkarsh Gaikwad (224101053)
3. Tejas Chandra Kareddula (224101052)

Contents

1. Acknowledgement 2

2. List of Figures 4

3. Abstract 5

4. Introduction 5

    4.1 What is Speech Recognition 5

    4.2 A small intro to our project 5

5. Experimental Setup 6

6. Proposed Techniques 7

    6.1 Flow chart 7

    6.2 Model Description 7

7. Result 8

    7.1 Home Page 8

    7.2 Live Training 8

    7.3 Live Testing 8

8. User Interface 9

    8.1 Initial UI 9

    8.2 Training Phase Interface 9

    8.3 Recording console 11

    8.4 Web page display 11

List of Figures

1. Flowchart of the project: . . . . . .

2. Home Page . . . . . .

3. Live Training . . . . . .

4. ExistingWord Search . . . . . .

5. New Word Search . . . . . .

6. Recording Console . . . . . .

7. Display of Webpage . . . . . .

1. Abstract

This project is developed using C++/C. It can take a speech sample of a few seconds, preferably a single word, and display the spoken corresponding product in online shopping sites which gives related information to that word. Initially it is developed for simple words for simplicity. But it can be expanded further for a bigger area of words. It uses the concepts of the famous Hidden Markov Model to store the properties of the speech sample and compare the new sample with these properties to detect which word has been spoken.

2. Introduction

2.1 What is Speech Recognition?

Speech Recognition is a technique which is quite popular now-a-days. When we speak into a micro- phone which is connected to the computer/mobile, it converts it to a text file which contains some amplitude values. Those values are basically the deviation of the speech signal from X-axis. Then we can use this file, do some calculations which can detect which word has been spoken and then further steps can be taken as per the requirement. One such application is Siri.

2.2 A Small Intro To Our Project

There is a set of predefined product names - IPhone, TV, Bat, mobile. We can run the project and speak any of these words to see the related product information in online shopping sites. We can add a new product name which might take 2-3 minutes to train the model.

3. Experimental setup

Basic requirements for this project are as follows-

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

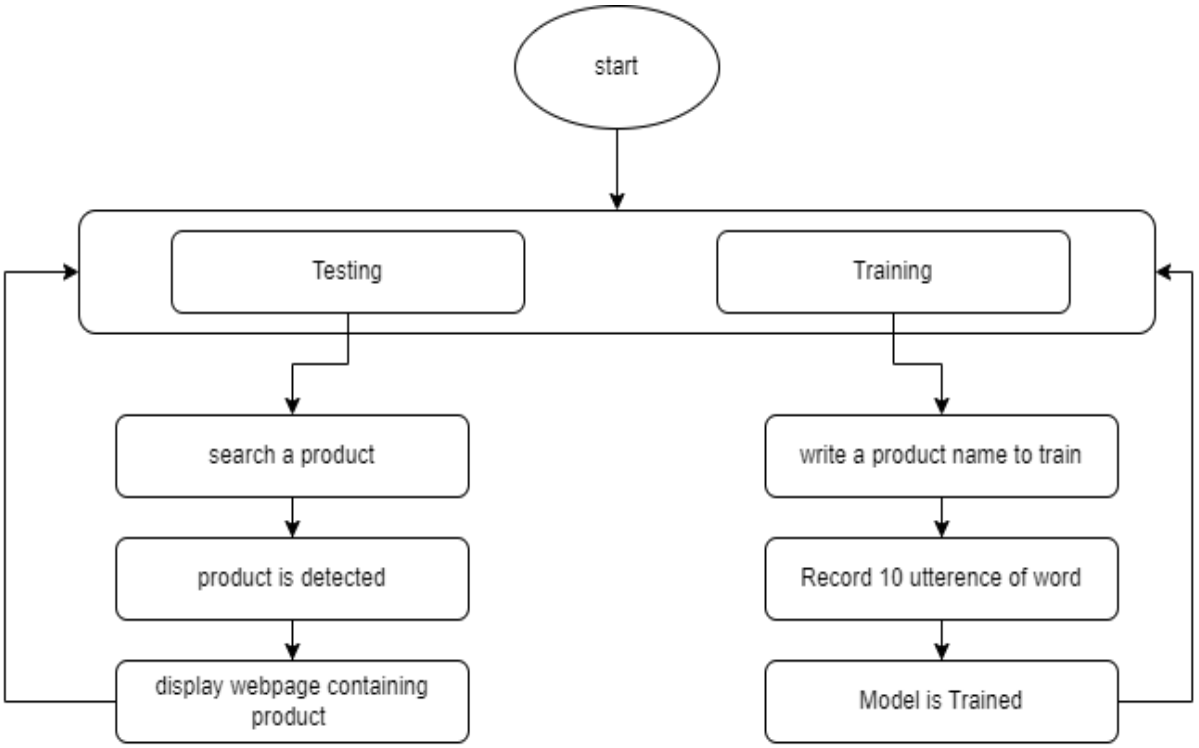

Figure 1: Flowchart of the project

4. Proposed Techniques

4.1 Flowchart

Figure 1 is a flowchart of this project. We can use these flowchart for the successful execution of our project.

4.2 Model description

We are utilizing the popular Hidden Markov Model to store the Speech Properties. Hidden Markov Model is a probabilistic model which is utilized to determine the probabilistic quality of any random process. At the point when we apply log function to the spectral representation of the speech during reverse Fourier Transform, it is switched over completely to cepstrum whose coefficients are consistent in view of utilization of log function and it addresses the discourse in a pleasant way. We take all such cepstral coefficients and generate a codebook which helps in creating the perception of successions. Codebook contains 10 speech samples for each word.

We use a feed-forward model for displaying speech samples. While talking, we express something from beginning to end. In this way, there is no need for reverse development. Likewise, the weight on the current phoneme is more than moving to the following phoneme. Consequently we use the feed-forward model. Then while testing, we score each model utilizing the forward cycle and pick the word with the most noteworthy score as the outcome.

Since, Speech signals rely a great deal upon the climate, live testing probably won't be generally excellent. Be that as it may, on the off chance that we train the model live and test it right away, we get essentially better exactness.

5. Result

5.1 Home Page

Home page contains two buttons: Live testing and Live training.

5.2 Live Training

We will find the word that is present in the Dictionary. If it is present it will show that it exists, else we will have to train the new word with 10 utterances, the UI will ask for the new word for the input. We need to wait for the codebook generation of the word

5.3 Live Testing

For live testing, a related button ought to be clicked. Then, at that point, a recording module will be opened. A word can be expressed and recorded for testing. The corresponding product will be displayed in the browser E commerce webpage from where we can purchase the product.

7. User Interface

7.1 Initial UI

Users will see two buttons here one is live test and another is Live train. So we choose accordingly

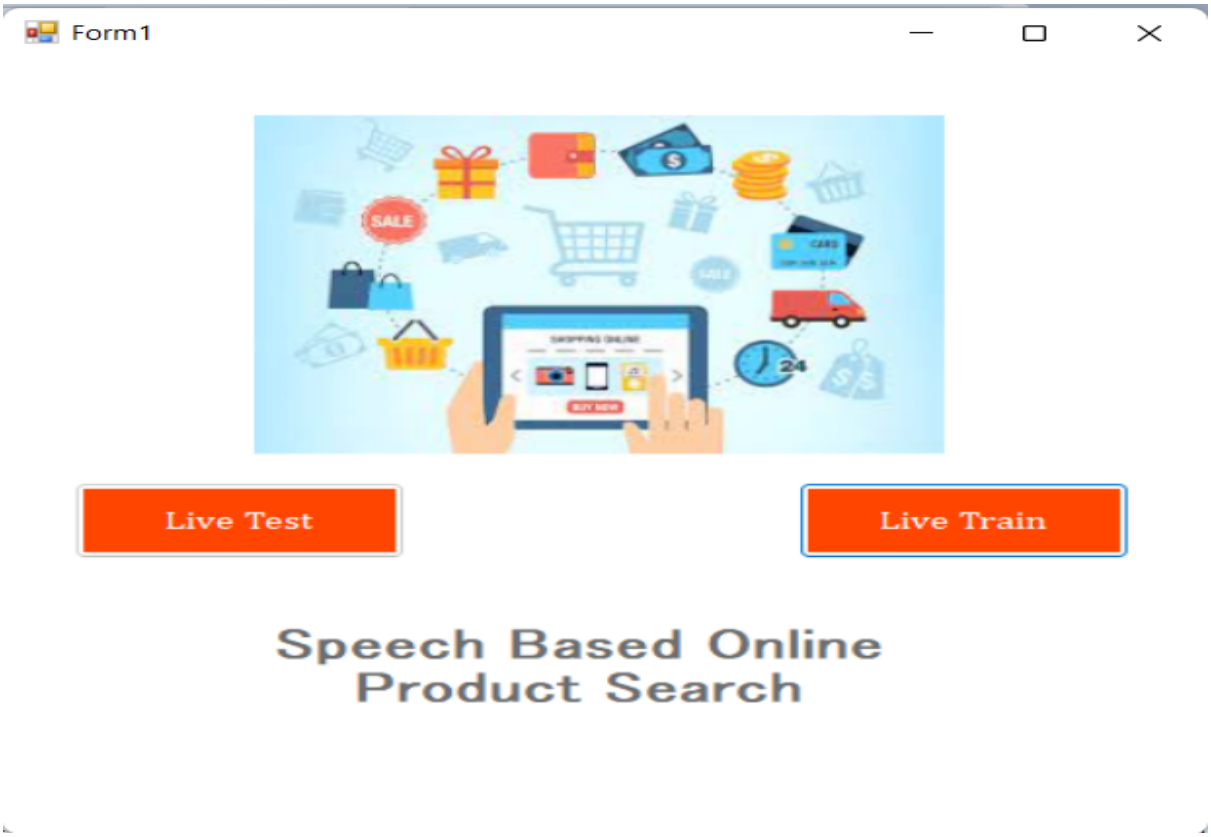

Figure 2

7.2 Training Phase Interface

We will find the word that is present in the Dictionary. If it is present it will show that it exists, else we will have to train the new word with 10 utterances, the UI will ask for the new word for the input. We need to wait for the codebook generation of the word.

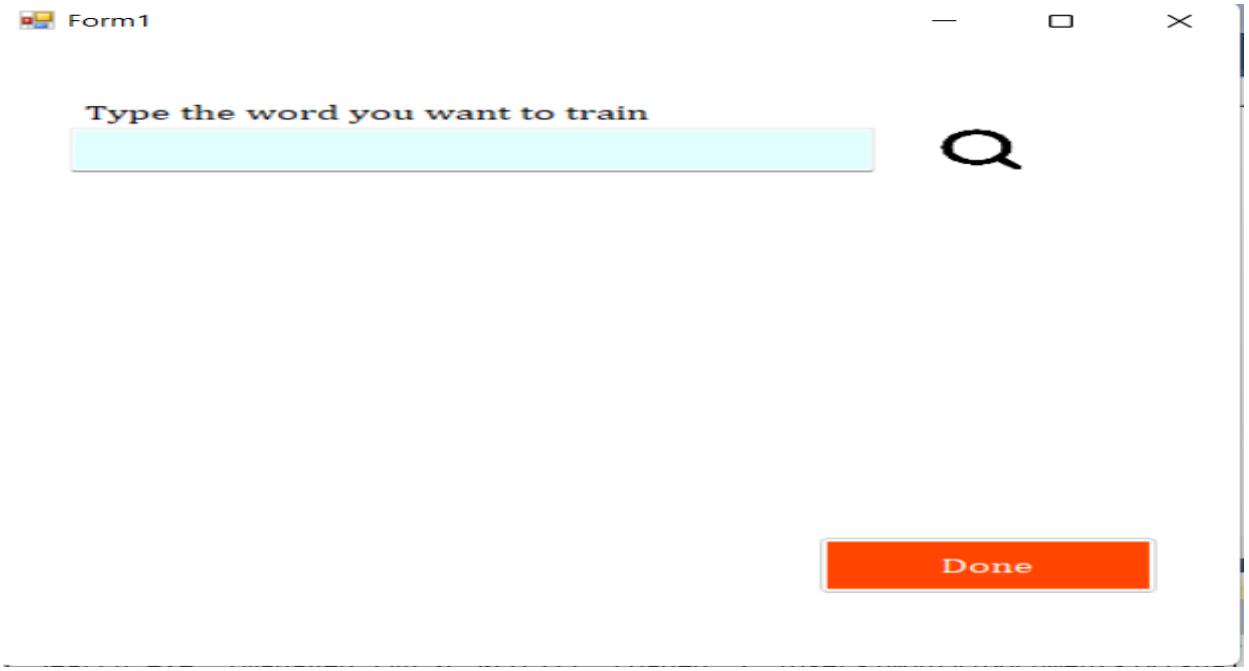

Figure 3

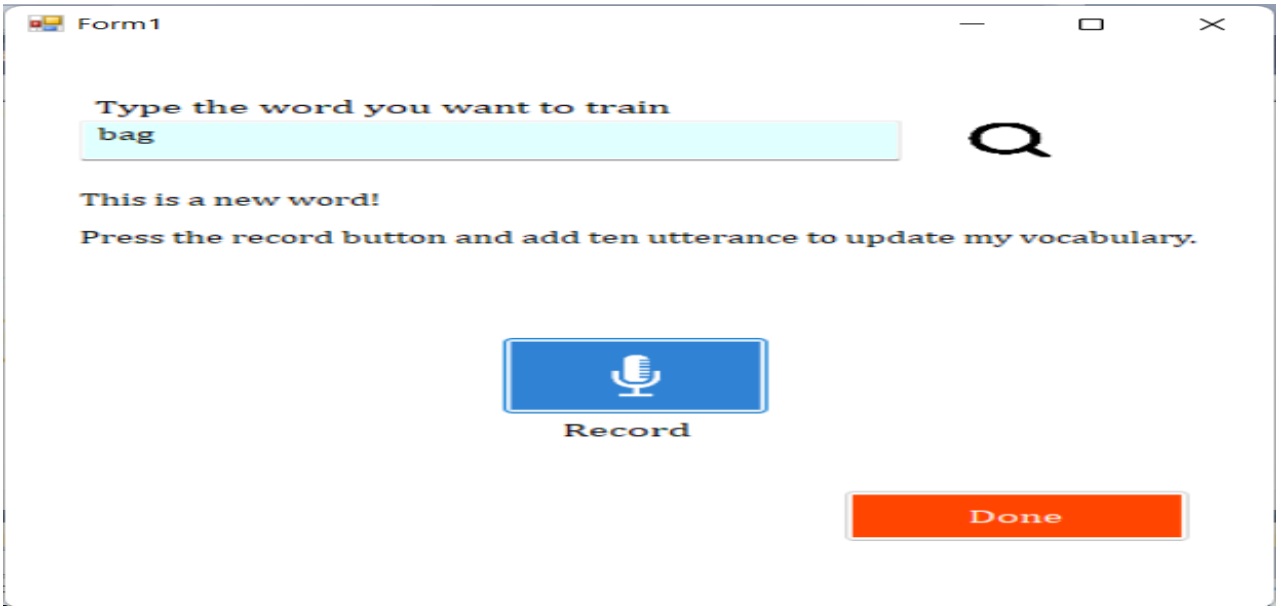

Figure 4

7.3 Recording console

In this console we will utter our word ten times. With start and end recording alerts.

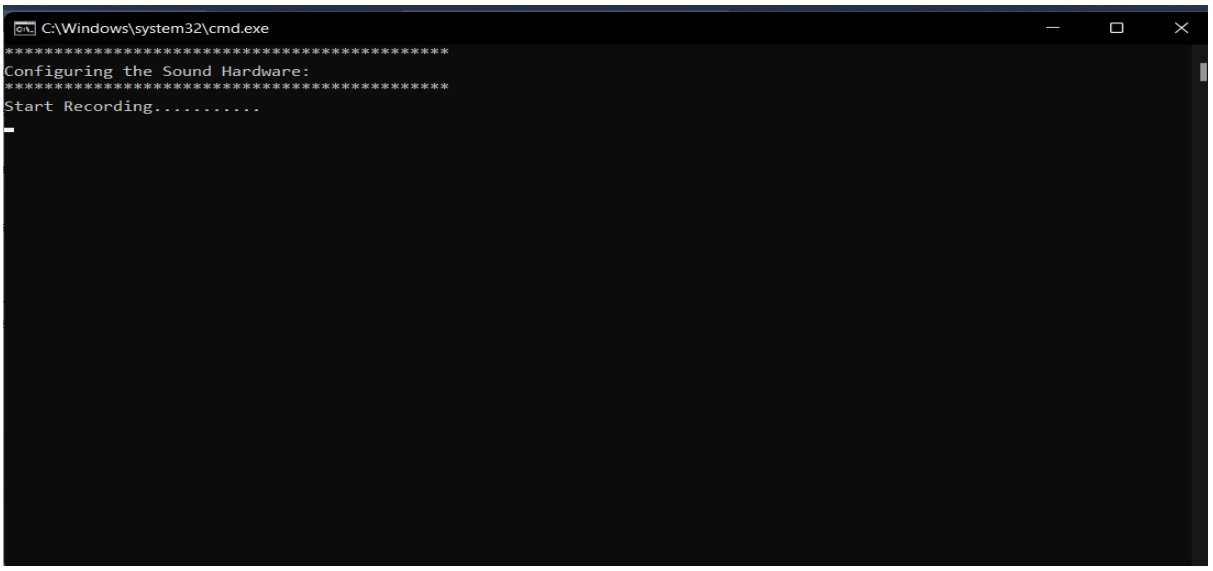

Figure 5

7.4 Web page Display

We will then test our model with the words we have, so if we make a successful test then the model will redirect it to the Webpage with the desired product name.

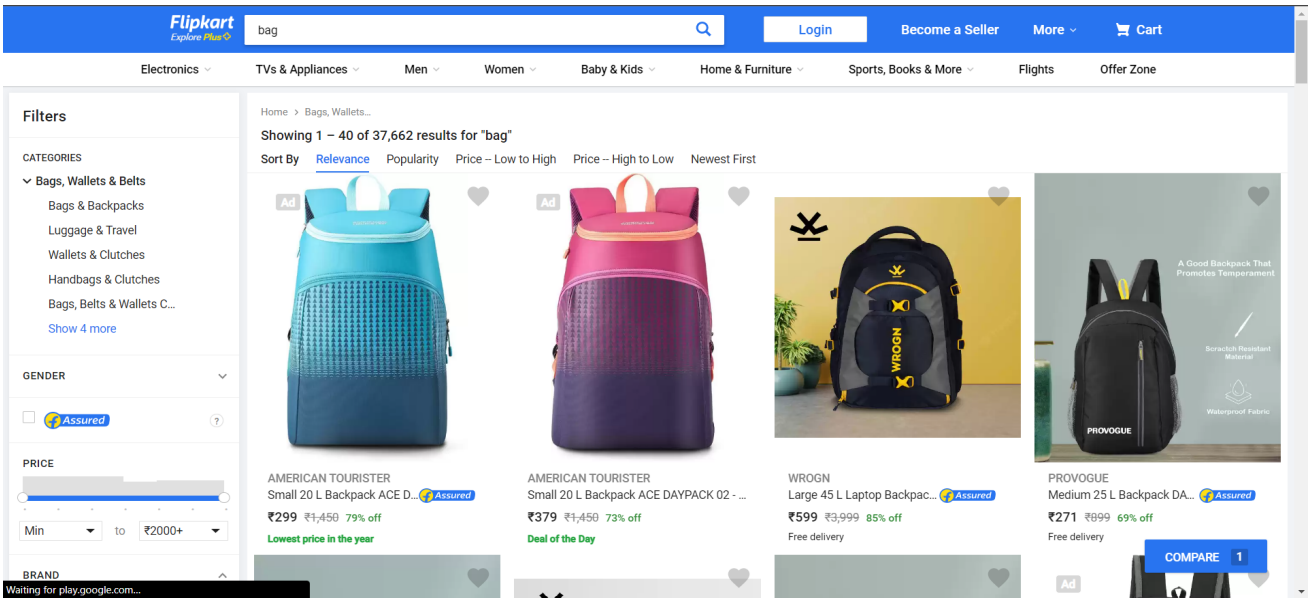

Figure 6