INDIAN INSTITUTE OF TECHNOLOGY, GUWAHATI

Department of Computer Science and Engineering

Project Report on

INTERACTIVE VOICE RESPOSE SYSTEM

Based on Speech Recognition System

Submitted to:

Prof. P. K. Das

Submitted by:

Kaja Gnana Prakash(224101027)

Ashish Kumar Pal(224101009)

Jaya Teja Reddy Pochimireddy(224101065)

For course fulfilment of CS566: Speech Processing

Acknowledgement

This project is being submitted as a requirement for course fulfilment 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. His timely guidance and suggestions were encouraging. We thank to the Teaching Assistants who were always helpful in clearing doubts. Finally, we thank to our classmates for the support.

1. Kaja Gnana Prakash (224101027)
2. Ashish Kumar Pal(224101009)
3. Jaya Teja Reddy Pochimireddy(224101065)

Contents

1 Abstract

This project is developed using C++/C. It can take a speech sample of a few seconds, preferably a single digit for required time, and based on user given digits corresponding Movie/show from Corresponding Genre will be played. But it can be expanded further by adding additional features like using words instead digits etc. 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 microphone 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 Alexa.

2.2 Our Project

This project uses a similar technique. It's a simple IVRS(Interactive Voice Response System) that Uses digits as input and based on that it provide service to the user which is movie selection. We can also train for new speakers. We can add numbers which might take several minutes.

2.3 Future improvements

Since this project is developed using C/C++, it is difficult to run multiple things parallelly. Future improvements include development of this project using High Level Languages like Java or Python which can use multithreading concepts to run the model training part in background. This will remove the waiting time during training of new features.

3 Experimental Setup

Basic requirements for this project are as follows-

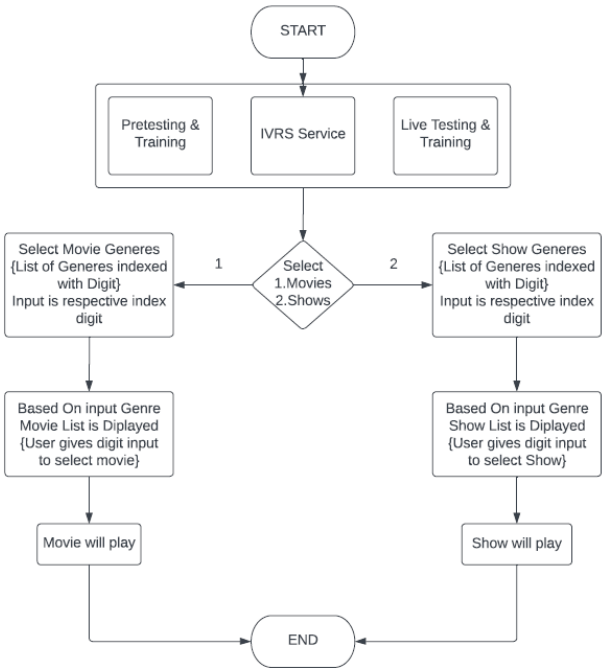

Figure 1: Flowchart of the project

- Windows OS
- Microsoft Visual Studio 2010
- C++11 integrated with VS2010
- Recording Module
- A good microphone

4 Proposed Techniques

4.1 Flowchart

Figure 1 is a flowchart of the project. Those steps can be followed for successful execution of the project.

4.2 Model description

We are using the famous Hidden Markov Model to store the speech properties. Hidden Markov Model is a probabilistic model which is used to explain or derive the probabilistic characteristic of any random

process. When we apply log function to the spectral representation of the speech during reverse fourier transform, it is converted to cepstrum whose coefficients are steady because of application of log function and it represents the speech is a nice manner. This representation can be used as the speech property. We take all such cepstral coefficients and build a codebook which helps in generating the observation sequences. Codebook contains 30 speech samples for each digit.

We use feed-forward model for modelling speech samples. While speaking, we speak a digit from start to end. So, there is no need of backward movement. Also, the stress on current phoneme is more than moving to the next phoneme. Hence we use feed-forward model. Then while testing, we score each model using the forward process and pick the word with highest score as the result.

Since, speech signals depend a lot on the environment, live testing might not be very good. But, if we train the model live and test it immediately, then we get significantly better accuracy.

5 Result

5.1 Home Page

Figure 1 shows the home page of the project. If Start IVRS button is clicked then Figure 5 will be displayed.

5.2 Live Testing

For live testing, corresponding button should be clicked. Then recording module will be opened which is showed in Figure 4. When IVRS Start Button is clicked then Figure 5 will displayed now user has to give digit input according to his requirement say for example Movies/shows --> Genres-->Selection

A sample example Walkthrough Figures 5-10

5.3 LiveTraining

For live training, corresponding button should be clicked. Then Figure 11 will be displayed. A Digit can be entered and the user has to give digit speech input for 20-30 utterances like wise the new user has to repeat for all digits after completing training then the model is ready to use for the new user.

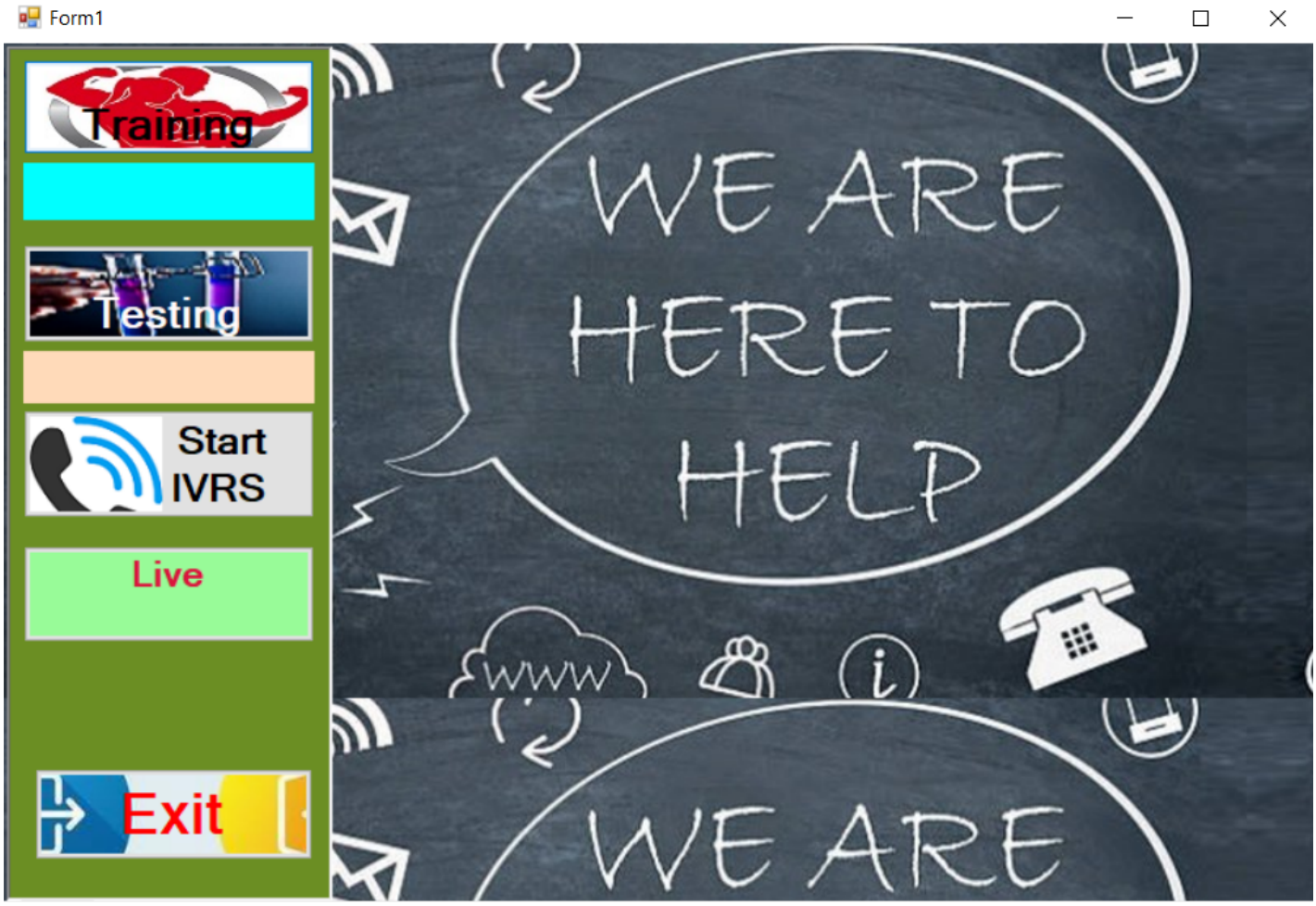

Figure 2: Home Page

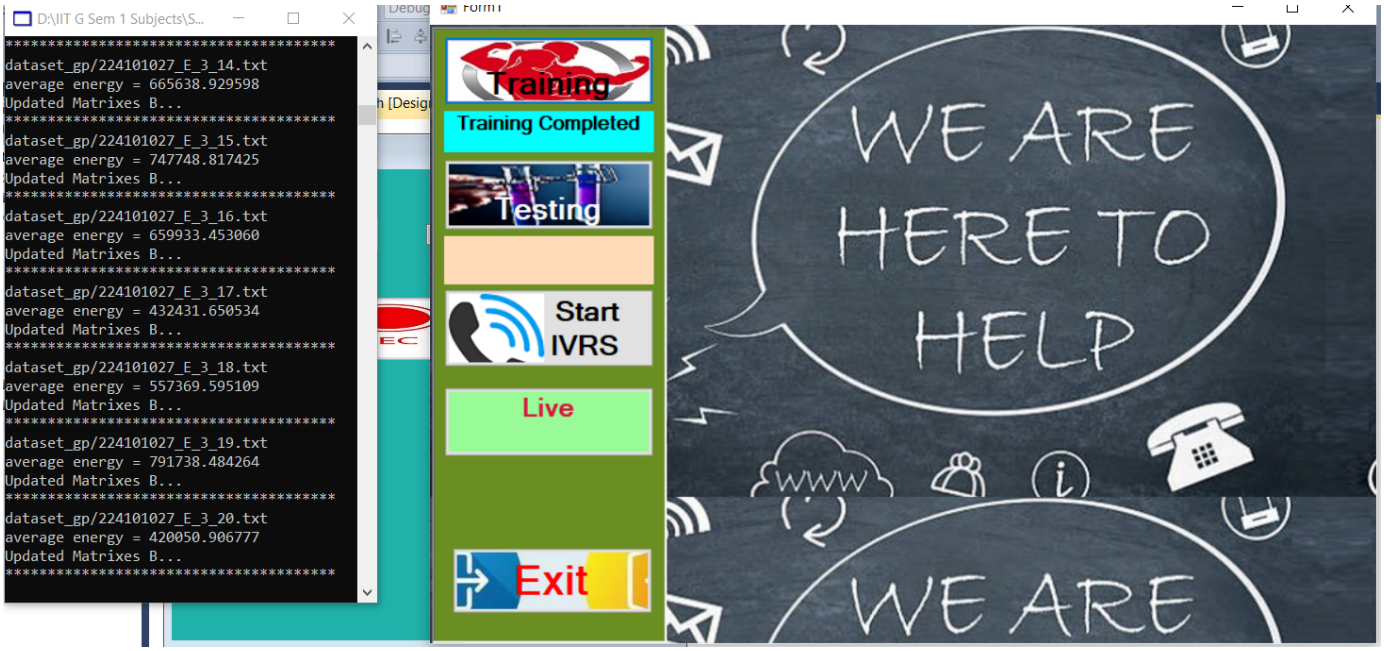

Figure 3: Training

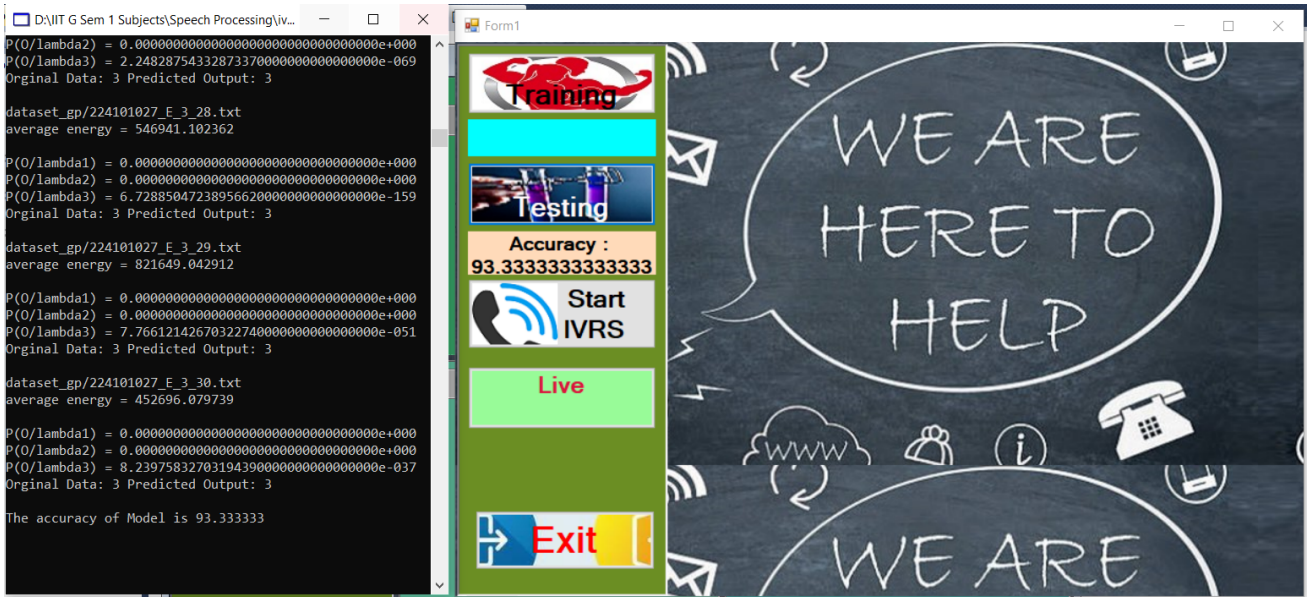

Figure 4: Testing

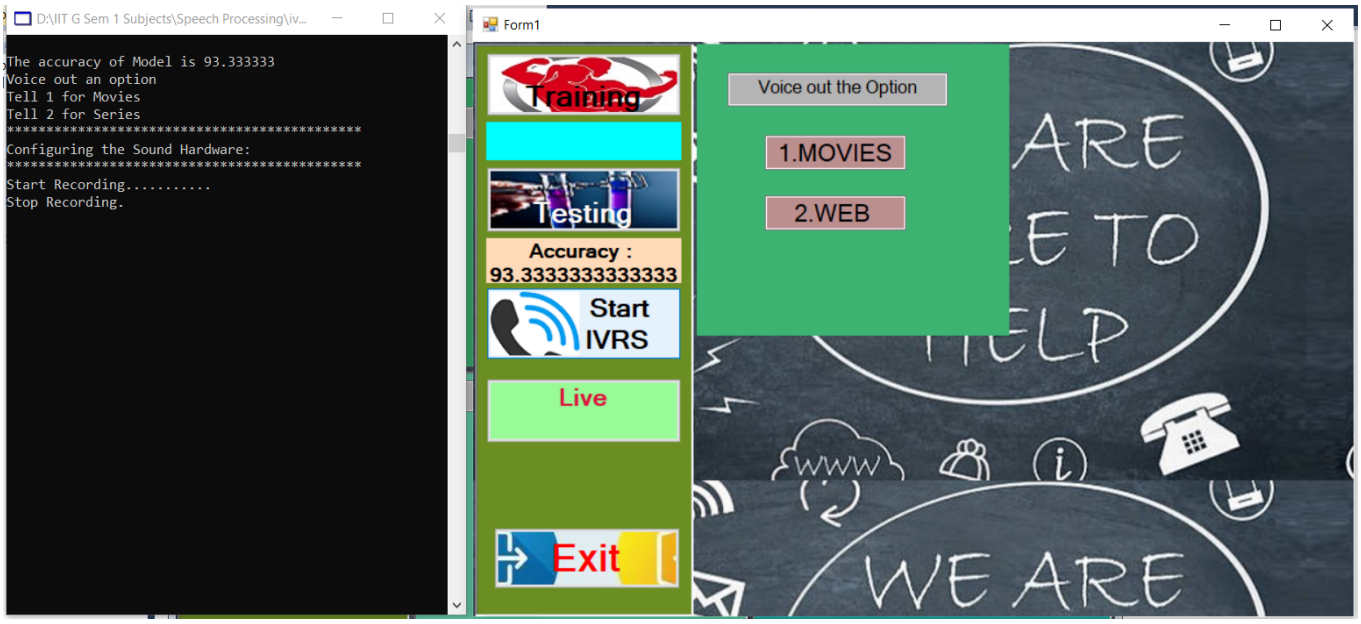

Figure 5: First Section of IVRS Movie/Show

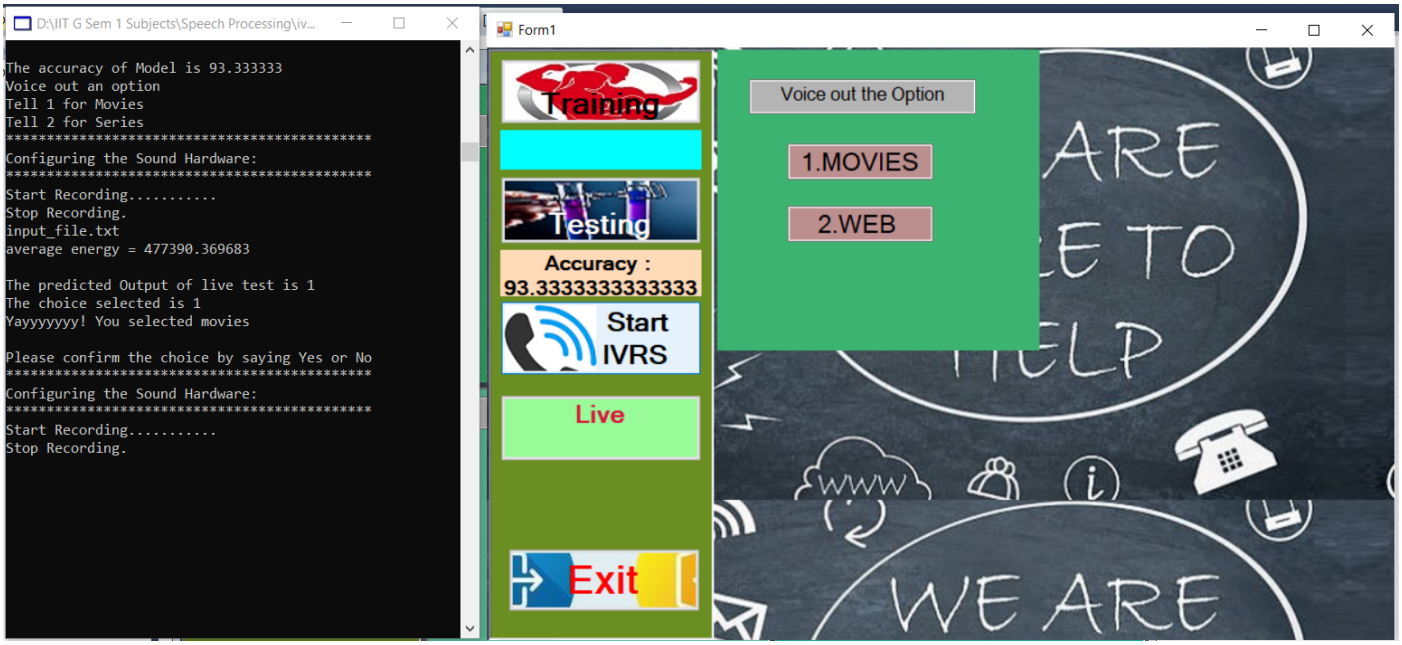

Figure 6: Section 1 Confirmation

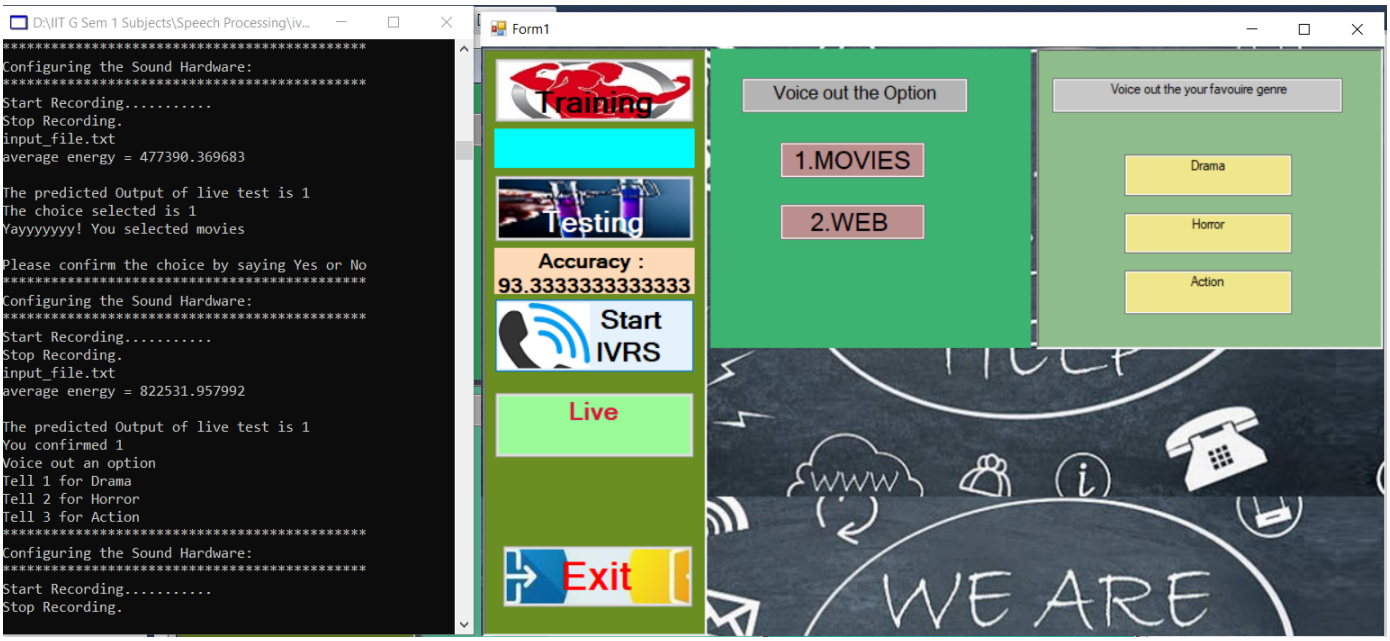

Figure 7: Section 2 Genre Selection

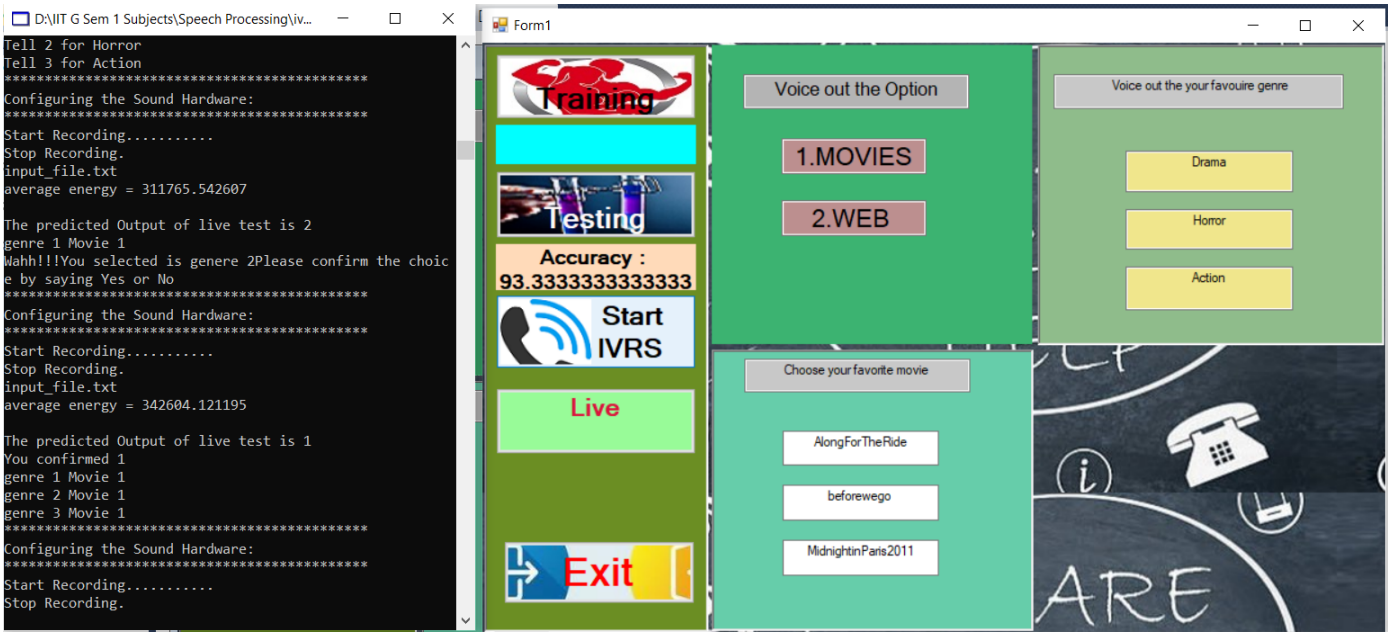

Figure 8: Section 3 Movie Selection

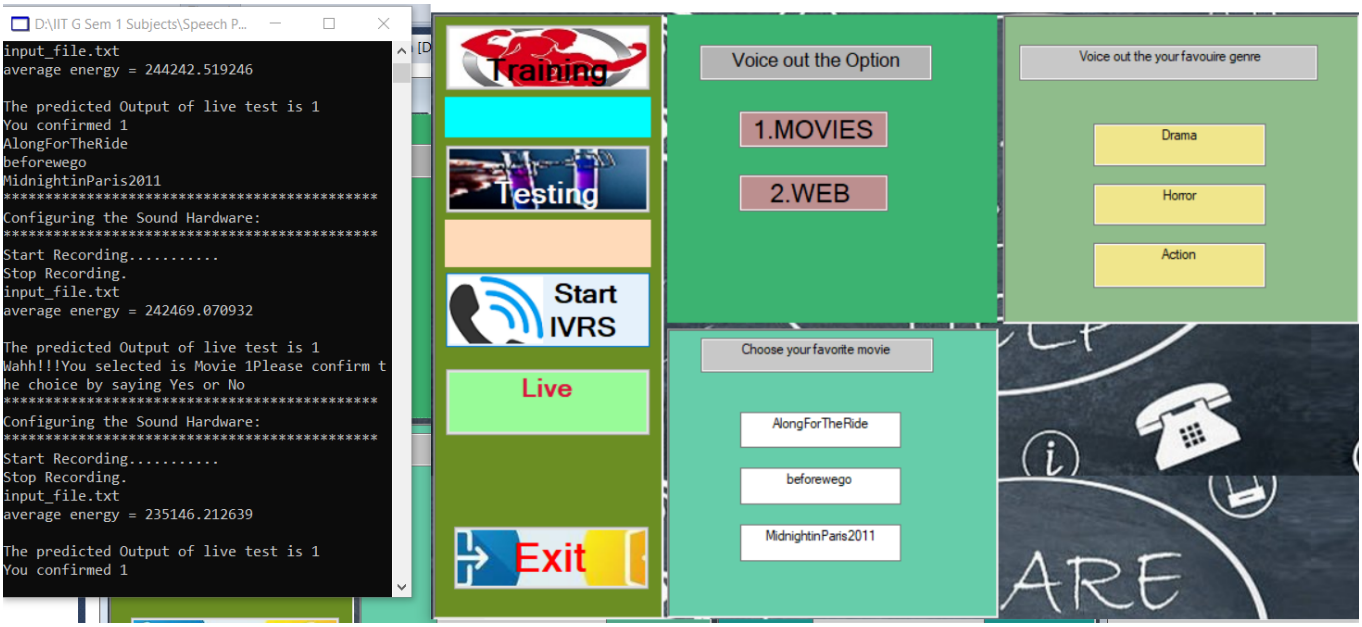

Figure 9 : Movie Confirmation

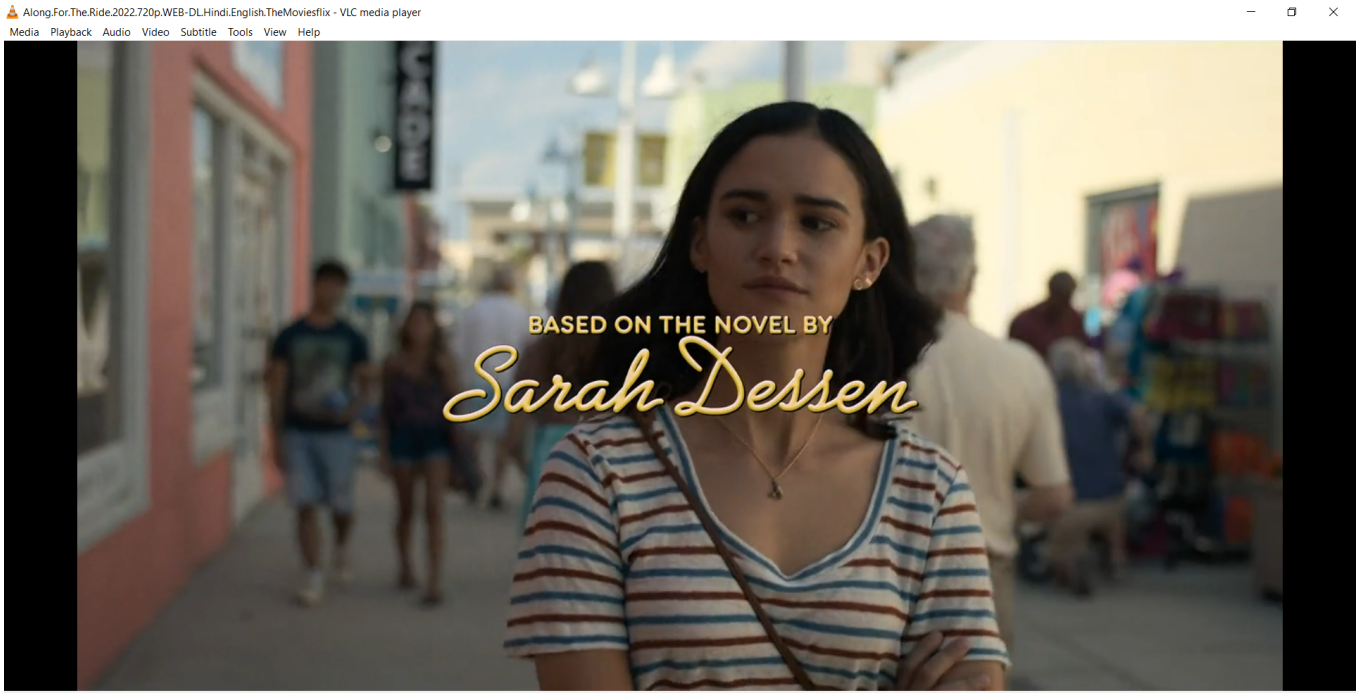

Figure 10: Section 4 Movie Play

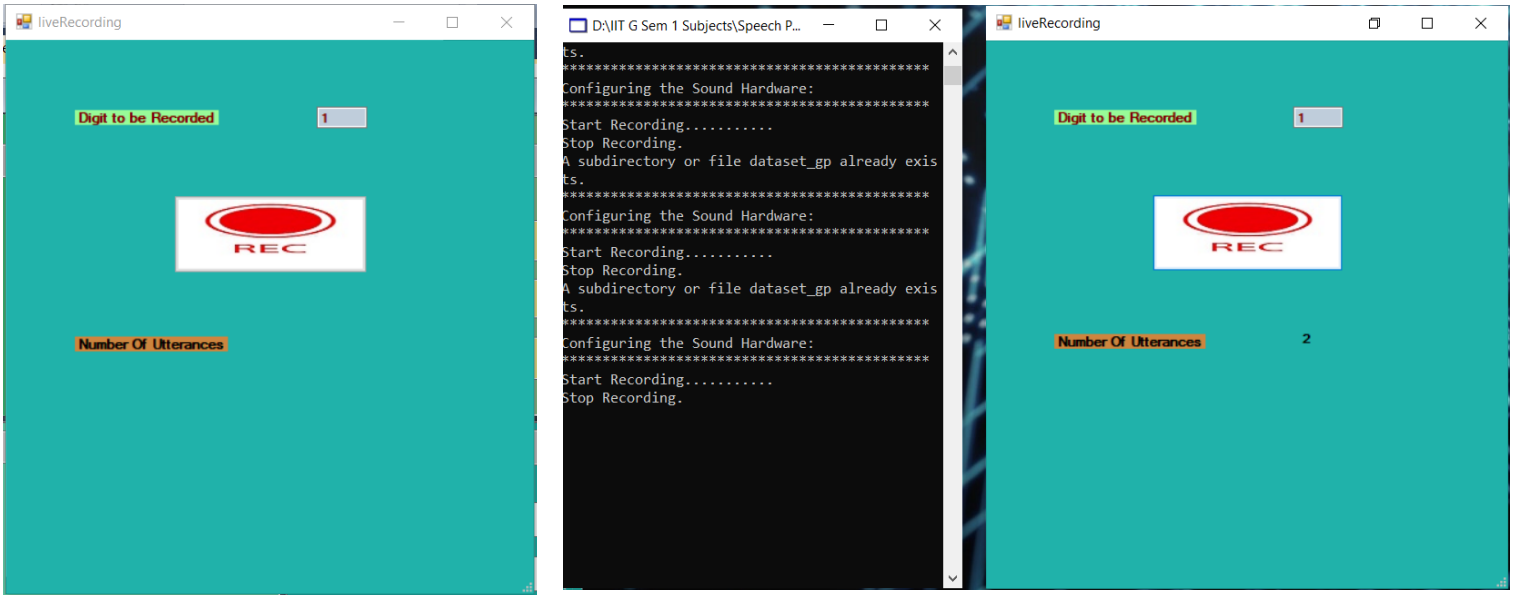

Figure 11 : Live Training