EAST WEST UNIVERSITY

EEE 309 Lab

(Open-Ended Lab Report)

Name of the Experiment: Open-ended lab to investigate characteristics of different commands/spoken words in Bangla and to design a simple Bangla speech recognition system

Date of Submission : 25.08.2023

Submitted to : Dr. Halima Begum
Assistant Professor
Dept. of Electrical and Electronic Engineering
East West University

Submitted by : Group 3

Group Members : Isteak Ahmed Khan Badhon (2020-1-80-007)
Sayeda Sayed Tasnuva (2020-1-80-045)
Meherin Islam (2020-1-80-048)
Shahjia Anjum (2020-1-80-063)
Md. Shariful Islam Raihan (2020-2-80-004)

Course Code : EEE309

Course Title : Digital Signal Processing

Section : 1

Name of the Experiment: Open-ended lab to investigate characteristics of different commands/spoken words in Bangla and to design a simple Bangla speech recognition system.

Objective:

The objective of this experiment is to analyze real audio signals (voice signals) and create a speech recognition system that can distinguish between the two audio signals "KOM" and "BESHI". This project also aims towards investigating several methods for analyzing Bangla audio signals.

Background information:

This project is designed as such so that we have clear knowledge about the mathematical idea of speech recognition. The main purpose of the procedure that we have adopted is to compare the mathematical value that we have gotten from our program and declare them as ‘Kom’ or ‘Beshi’ according to our chosen train data. To implement this program, we chose MATLAB 2022a and 2022b. We have around 300 Train data with the same sampling Frequency (44.1 KHz) . Among these 300 train data, we chose those train data that were able to show the maximum accuracy. Even though our program was able to show 100% accuracy, this will not work out on every new test data. There can be a chance of getting wrong detection in some cases. This is why we observed multiple methods to get the maximum result.

Literature Review:

1. **Bangla Short Speech Commands Recognition Using Convolutional Neural Networks**

   In this research a convolutional neural network (CNN) based architecture for Bangla short speech recognition has been applied. In their model the traditional CNN consists of layers stacked together which are an input layer, a group of convolutional and pooling layers, several fully connected layers, and finally an output layer. The convolutional and pooling layers, followed by fully connected layers are the main differences of CNN compared to other neural networks, and this kind of special layer architecture has significant practical consequences in terms of speech recognition.

They have collected utterances of 10 different words of a total 65,000 samples in real life noisy conditions and trained CNN-based models on them. This system showed the highest 85.44% accuracy. [1]

2. **A Speech Recognition System for Bengali Language using Recurrent Neural Network:**

   This paper implemented a speech recognition system in Bengali language by using two neural networks, namely, convolution neural network and recurrent neural network. The CNN model could convert isolated speech signals into texts and had an accuracy of 86.058%. This model was trained with 30,000 Bengali words.[2]

3. **Bangla Speech Recognition System Using LPC and ANN**

   In this research the system worked in two parts. The first part is speech signal processing and the second part is speech pattern recognition technique. The speech processing stage consists of speech starting and end point detection, windowing, filtering, calculating the Linear Predictive Coding(LPC) and Cepstral Coefficients and finally constructing the codebook by vector quantization. The second part consists of a pattern recognition system using the Artificial Neural Network(ANN). [3]

Methodologies on Human Voice Recognition:

1. **Cross-correlation Method:** Cross-correlation is used to align the input speech signal with a reference signal. By sliding the reference signal over the input speech signal and calculating the similarity at each position, cross-correlation helps identify the occurrence of the reference signal or specific reference points in the input audio. The time shift with the highest correlation value indicates the optimal alignment between the two signals, allowing voice recognition systems to accurately detect and recognize keywords or commands within the spoken input. This alignment process is crucial for efficient and accurate speech recognition in applications like virtual assistants and voice-controlled devices.

2. **Linear Predictive Coding (LPC):** Linear Predictive Coding (LPC) is used for speech analysis. It approximates speech by modeling the spectral envelope using linear predictive coefficients. LPC extracts crucial features like Mel-frequency cepstral coefficients (MFCCs) from speech frames, which are then used to recognize phonemes, words, or commands in the voice recognition system. LPC's ability to compactly represent speech and capture spectral characteristics makes it valuable for voice analysis and feature extraction in speech recognition applications.
3. **WFST-based Decoding:** It efficiently searches through a large set of possible transcriptions using Weighted Finite State Transducers (WFSTs). Language models, acoustic models, and pronunciation dictionaries are represented as WFSTs and combined into a single network. The search algorithm explores different paths in the composite WFST to find the most likely sequence of words or phonemes, representing the recognized speech. WFST-based decoding is valuable for handling large vocabularies and real-time speech recognition applications.
4. **Frequency Domain Analysis:** Frequency domain analysis in voice recognition plays a critical role in understanding and processing speech signals. It involves transforming audio signals from the time domain to the frequency domain using mathematical techniques like the Fourier Transform. By doing this, it decomposes the signal into its constituent frequencies, enabling the extraction of vital frequency-based features like Mel-frequency cepstral coefficients (MFCCs). MFCCs represent the spectral characteristics of speech and are widely used as inputs for voice recognition systems. Spectrograms, a visual representation of the frequency content over time, offer valuable insights into how the speech signal's spectrum changes during different intervals. This analysis helps in pitch detection, which is crucial for identifying prosody and intonation in speech. Furthermore, frequency domain techniques facilitate noise reduction by filtering out unwanted frequency components, thus improving the signal quality before voice recognition processing. Additionally, spectral enhancements can be applied to emphasize important frequency regions, leading to better voice recognition performance in noisy environments. Overall, frequency domain analysis enhances voice recognition by providing a comprehensive understanding of the spectral properties of speech, enabling the extraction of critical features, and improving the system's accuracy and robustness

5. **Normalized Cross-correlation:** Normalized Cross-correlation is a method used to align a reference signal with an input speech signal to identify occurrences or specific reference points. It measures the similarity between the two signals as the reference signal is shifted relative to the input speech. The cross-correlation values are normalized to ensure the similarity measure is independent of the signal amplitudes. The time shift with the highest normalized cross-correlation value indicates the optimal alignment, helping accurately detect and recognize keywords or commands within the spoken input, enhancing voice recognition performance.

Preferred Method: The method we have chosen is 'Normalized Cross-Correlation'. It is a mathematical tool or statistical measure that we use to analyze the similarities and dissimilarities between two signals or two sets of data. The value of this process varies from +1 to -1. When the value is closer +1,it indicates that the two signals are more similar in nature. On the other hand, when the value is closer to -1, it indicates that the signals are more dissimilar in nature. 'Normalized Cross-Correlation' process is much easier to use because it gets the job done of comparing different signals simply. This process is generally used in case of finding similarity or dissimilarity, recognizing pattern etc. Also, it makes the whole process more interpretable based on its range of value which differs from +1 to -1.

Observation: Our test data were 'Kom' and 'Beshi'. If we look at the first two normalized cross correlation figure (Figure-1 and Figure-2), we can see that when we did the normalized correlation between 'Kom' test data and 'Kom' train data the amplitude =0.197455 was higher than the amplitude =0.0621007 of the normalized correlation between 'Kom' test data and 'Beshi' train data. As the amplitude was higher in the first case, our program detected it as 'Kom'

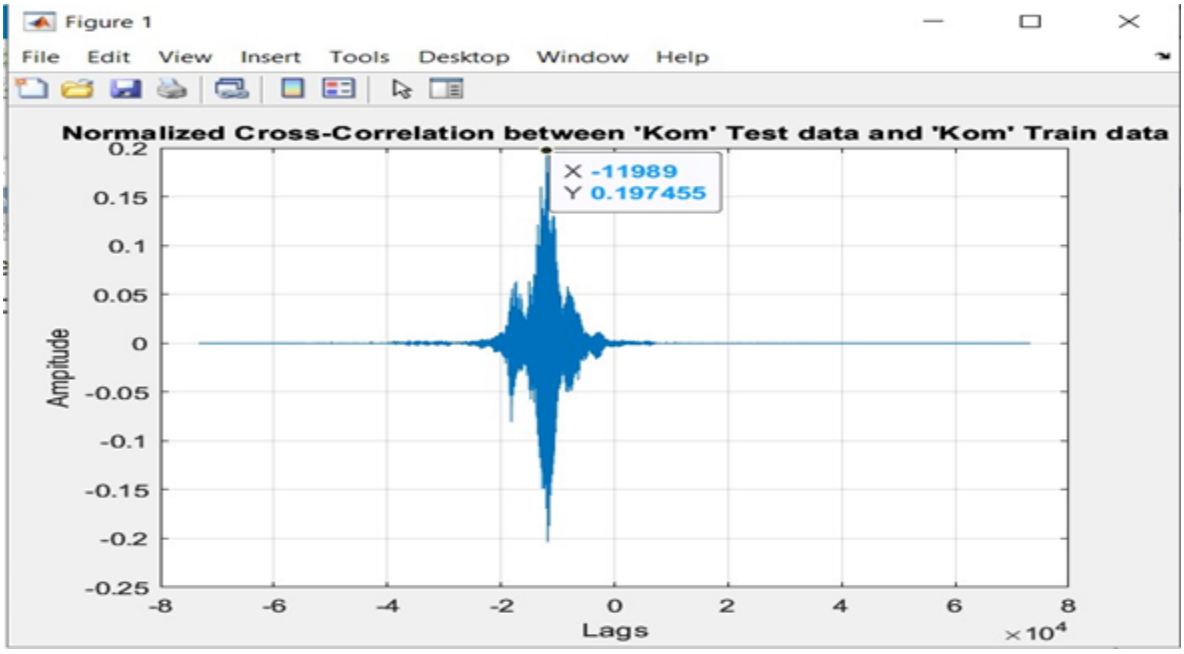


Figure-1: Normalized Cross-Correlation between 'Kom' Test data and 'Kom' Train data.

In Figure-1 we did normalized cross-correlation between test data T-1 and train data L-95.

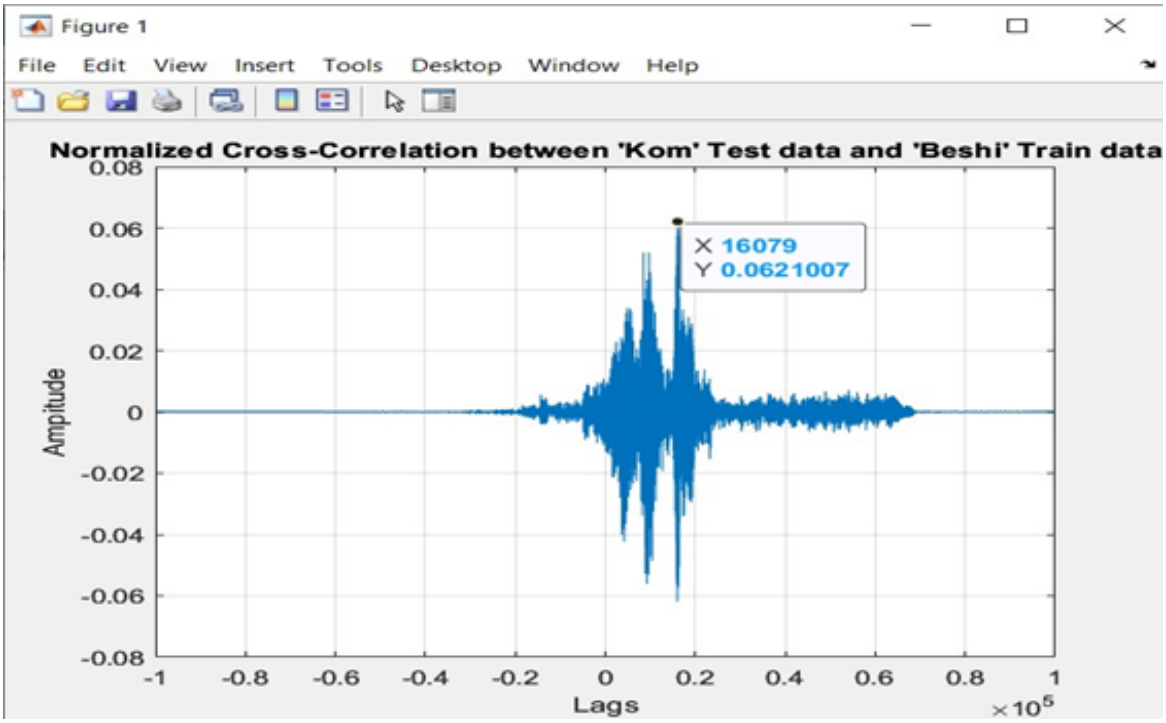


Figure-2: Normalized Cross-Correlation between 'Kom' Test data and ‘Beshi’' Train dataIn

Figure-2 we did normalized cross-correlation between test data T-1 and train data M-26.

Now, if we look at the last two normalized correlation figures (Figure-3 and Figure-4) below, we can say that when we did the normalized correlation between ‘Beshi’ test data and ‘Kom’ train data the amplitude =0.0447589 is lower than the amplitude=0.0680309 of the normalized correlation between ‘Beshi’ test data and ’Beshi’ train data. As the amplitude is higher in the second case, our program detected it as ‘Beshi’

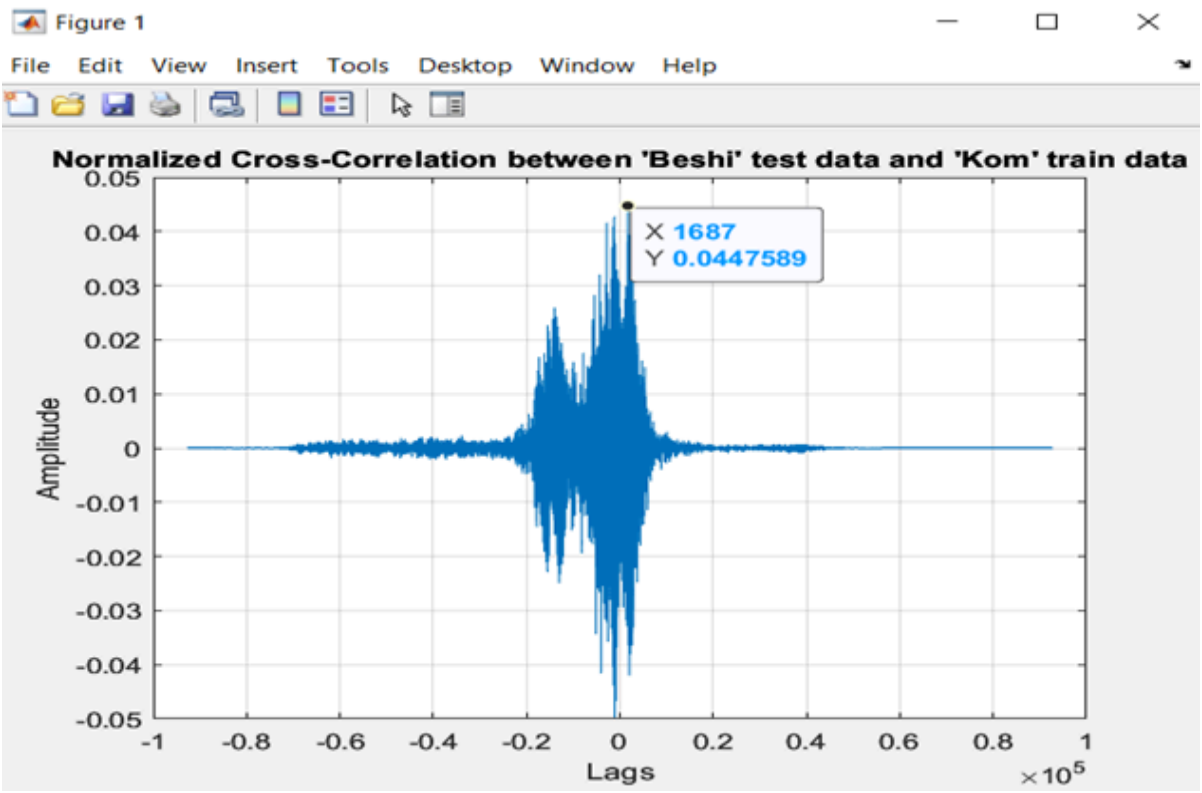


Figure-3: Normalized Cross-Correlation between 'Beshi' Test data and 'Kom' Train data

In Figure-3 we did normalized cross-correlation between test data T-5 and train data L-40

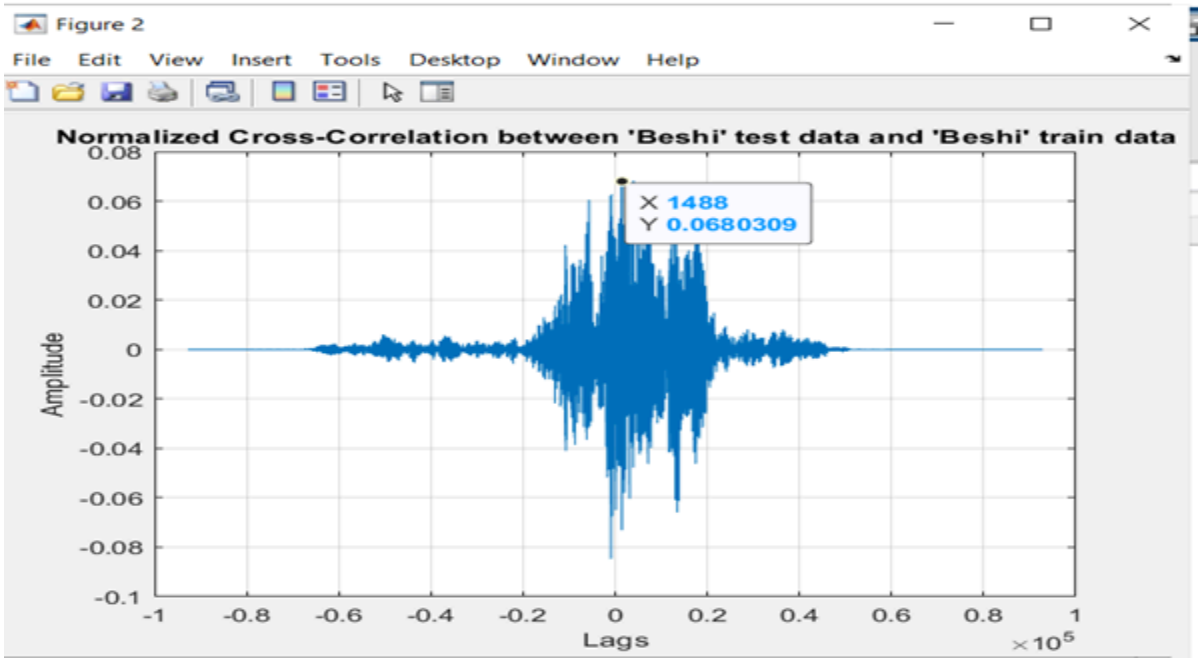
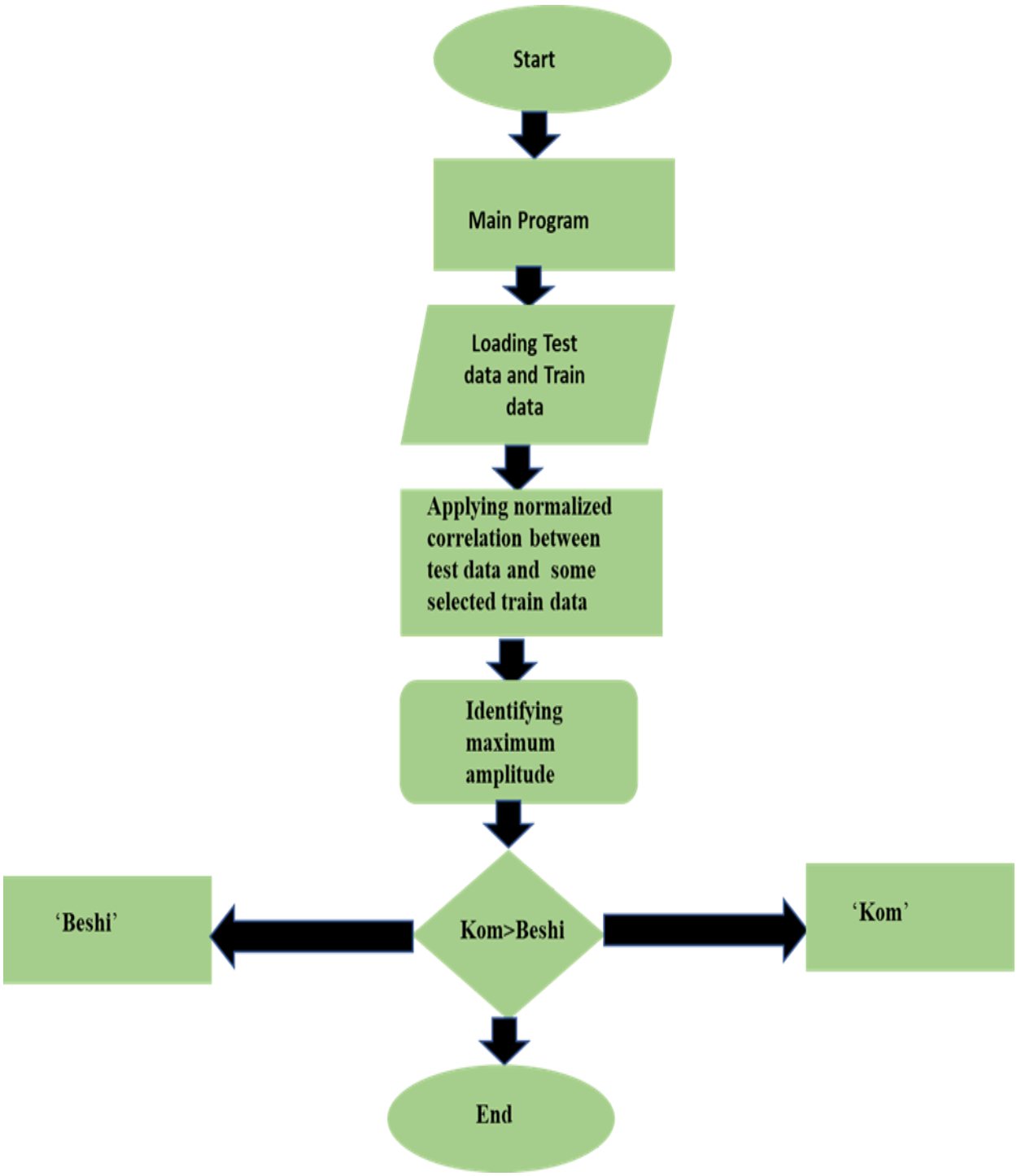


Figure-4: Normalized Cross-Correlation between 'Beshi' Test data and 'Beshi' Train data

In Figure-4 we did normalized cross-correlation between test data T-7 and train data M-103.

Our all test data from T-1 to T-10 showed this similar nature with a total of sixteen train data. Among those 16 train data, 8 data were from ‘Kom’ and other 8 data were from ‘Beshi’. Those train data were able to distinguish the test data accurately based on their maximum amplitude.

Flowchart:


Start
Main Program
Loading Test data and Train data
Applying normalized correlation between test data and some selected train data
Identifying maximum amplitude
Kom>Beshi
'Beshi'
'Kom'
End

Procedure:

1.At first, we read the audio files by using audioread(). Test data and some selected train data(both kom and Beshi) were read simultaneously.

2.After reading the files, we did normalized cross-correlation between the test data and both train data.

3.Then we used the if else condition.

```
if max(norm_l)>max(norm_m);

disp('kom')

else

disp('Beshi')
```

4. By applying this logic, our train data will be able to detect the test data accurately

Procedure to choose train data: We have selected a total of sixteen train data out of 300 train data. Among these sixteen train data, eight of them were from 'Kom' train data and the other eight of them were from 'Beshi' train data. We selected these train data based on the value of the maximum amplitude of the normalized correlation between test data and train data. For example, we fixed L-1 and M-1 then we kept testifying the test data one by one using a normalized correlation method. If our test data was 'kom', and the normalized correlation value between L-1 and test data was higher than the value of normalized correlation between M-1 and test data, the result was considered as 'Kom' otherwise it was 'Beshi'. We kept changing the test data one by one. If L-1 and M-1 were able to detect all 10 of these test data we selected L-1 and M-1 as our desired train data. The same procedure was repeated for train data from (L-2 and M-2 to L-150 and M-150). This is how we selected our sixteen train data.

Result and Discussion:

Result:

Number of the test data	Actual result	Detected Result	Result
1	Kom	Beshi	**Wrong**
2	Beshi	Beshi	Correct
3	Beshi	Beshi	Correct
4	Beshi	Beshi	Correct
5	Kom	Kom	Correct
6	Kom	Kom	Correct
7	Kom	Kom	Correct
8	Beshi	Beshi	Correct
9	Kom	Beshi	**Wrong**
10	Beshi	Beshi	Correct

Comment:

Our system has successfully identified 8 out of 10 data.It can detect all the " Beshi" Data correctly but it could not detect 2 "Kom" Data correctly.

Discussion: Our system showed an 80% accuracy on the final demonstration.We observed that it was difficult to detect ‘Kom’ test data for some reason. We analyzed the value of the signal to noise ratio of both types of test data and we found out that the signal to noise ratio of ‘kom’ test data is significantly higher than the ‘Beshi’ test data. This signal to noise ratio is what made the difference. This proved that the ‘Kom’ test data were noisier than ‘Beshi’ test data. To sum up, in the case of ‘kom’ test data the signal power is lower than the noise power. This is why our program failed to detect T1 and T9 .

Suggestion for Better Result: To improve the outcome, filtering out the noise is one of the options we can follow. Filtering or canceling out the noise from these test data will give us the maximum accuracy that we want.

Learning from the process: From this procedure we learnt that, only finding out the solution is not enough. There can be obstacles to finding out the actual result. From finding out the obstacle to sorting them out, we have to make clear and concise decisions. But this is not only limited to taking decisions. Acting upon the decision is another important factor that we have to consider. So, we can say finding out the problem and implementing the solution are the major factors that we got to learn from this experience.

Conclusion: In this experiment, we used MATLAB for analyzing both train data and test data. Throughout this experiment, we grasped theoretical knowledge, mathematical knowledge and programming knowledge. This also gave us the experience of tinkering with different approaches. To sum up, this Experiment was very informative which gave us clear concepts about Bangla speech recognition systems. This experiment will be helpful and beneficial for us in future if we work with similar types of projects.

References:

[1] Sumon, S. A., Chowdhury, J., Debnath, S., Mohammed, N., & Momen, S. (2018, September), "Bangla short speech commands recognition using convolutional neural networks," in *2018 international conference on bangla speech and language processing (ICBSLP)* (pp. 1-6).

[2] Islam, J., Mubassira, M., Islam, M. R., & Das, A. K. (2019, February), "A speech recognition system for Bengali language using recurrent neural network," in *2019 IEEE 4th international conference on computer and communication systems (ICCCS)* (pp. 73-76).

[3] Paul, A. K., Das, D., & Kamal, M. M. (2009, February), "Bangla speech recognition system using LPC and ANN," in *2009 seventh international conference on advances in pattern recognition* (pp. 171-174).

Appendix:

Main code:

```
clc
clear
close all
files1 = dir('L*.mp3');
names1 = {files1.name};
files2 = dir('M*.mp3');
names2 = {files2.name};
[test,fs] = audioread("T-10.mp3");
sound(test,fs)
x=test(:,1);
for i=1:8
[K{i},fs1{i}] = audioread(names1{i});
x1 = (K{i}(:,1));
[B{i},fs2{i}] = audioread(names2{i});
x2 = (B{i}(:,1));
c1{i} =(xcorr((x1),(x)));
c1_n = (c1{i}/sqrt(sum(x.^2).*(sum(x1.*x1))));
s1(i) = max(c1_n);
c2{i} = (xcorr((x2),(x)));
c2_n = (c2{i}/sqrt(sum(x.^2).*(sum(x2.*x2))));
s2(i) = max(c2_n);
end
kom = max(s1);
```

```
Beshi = max(s2);

if kom>Beshi

disp('Kom');

else

disp('Beshi');

end
```