BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Electrical and Electronic Engineering

Course No. : EEE 312 sessional
Course Title : Digital Signal Processing I Laboratory **Section :** B1

Project Report

Submitted to:
Shahed Ahmed
Lecturer, Department of Electrical and Electronic Engineering,
Bangladesh University of Engineering and Technology.
Barproda Halder
Lecturer(Part-Time), Department of Electrical and Electronic Engineering,
Bangladesh University of Engineering and Technology.

Submitted by:
1906078: Asif Aftab Ronggon
1906079: Md. Faiyaz Abid
1906080: Md. Liton Ali
1906081: Anindya Kishore Chowdhury

Table of Contents

Objective

- Keyword identification in MATLAB
- Detect foreign keywords and identify them as "none"

Problem Statement

The keywords we chose are: "fever", "cough", "medicine", "tablet", "syrup", "pain". We have to create a MATLAB program such that it can identify these keywords if one of them is spoken in front of microphone. Also, if a foreign keyword is spoken (which is not included in our keywords dataset), the program should display "none". This was our bonus problem.

Methodology

We performed the following tasks:

1. Collecting Samples
2. Creating dataset
3. Performing MFCC on all audio files
4. Recording speaker's keyword
5. Performing DTW between every dataset and recorded keyword
6. Showing result

Collecting Samples:

Voice samples were collected reaching on-person. We collected about 83 samples for each keyword. After collecting samples, we then sorted out the better samples and kept those for use.

Creating Dataset:

Dataset for each word was created using following code:

```
% set the path to the main directory
mainDir = 'D:\DSP PROJECT_174';
% define the labels and subdirectories
```

```
labels = {'cough', 'fever', 'foreign','medicine', 'pain', 'syrup', 'tablet'};
subDirs = {'cough', 'fever','foreign', 'medicine', 'pain', 'syrup',
'tablet'};


for i = 1:length(labels)
    % set the path to the subdirectory
    subDirPath = fullfile(mainDir, subDirs{i});
       % list the WAV files in the subdirectory
    wavFiles = dir(fullfile(subDirPath, '*.wav'));

    % loop through each WAV file and extract its feature
    a=[];
    for j = 1:length(wavFiles)
        % read in the audio data
        [audio_data, sample_rate] = audioread(fullfile(wavFiles(j).folder,
wavFiles(j).name));

        a=[a audio_data];
    end
  save(strcat('audiodata_',string(labels(i))),"a");

end
```

Performing MFCC on all audio files

The process of extracting MFCC features involves several steps. First, pre-emphasis is done by filtering the high frequency components. Then the signal is segmented into frames by applying a windowing function. Next, the Discrete Fourier Transform (DFT) is applied to each frame to obtain the frequency spectrum. The magnitude of the spectrum is then converted to a logarithmic scale. To account for the non-linear nature of human hearing, the frequency axis is warped on a Mel scale. Finally, the inverse Discrete Cosine Transform (DCT) is applied to obtain the final MFCC features. [1] Here, we used 13 MFCC features, the first feature is ignored. Code for performing MFCC is given:

```
%% Compute MFCC features for all audio files
cz=mfcc(audio_data(:,1,1),44100,"LogEnergy","replace");
cz(:,1)=[];
[r,c]=size(cz);
mfc_features=zeros(r,c,i,6);

for i = 1:length(labels)
    for j = 1:num_files
        M_temp=mfcc(audio_data(:,j,i), 44100, "LogEnergy", "Replace");
        M_temp(:,1)=[];
        mfc_features(:,:,j,i)=M_temp;
    end
end
```

Recording speaker's keyword

Recording was done using the following code:

```
%% Record a new audio sample and compute its MFCC features
Fs = 44100;
nBits = 16;
nChannels = 1;
device = 0;
recObj = audiorecorder(Fs, nBits, nChannels, device);
recTime = 2;
disp('Start Speaking ');
recordblocking(recObj, recTime);
disp('Recording Stopped');
disp('Stop speaking!!!');
```

Performing DTW between every dataset and recorded keyword

Dynamic Time Warping (DTW) is a signal matching method where two signals are compared and their similarities are computed. Here, we performed MFCC in our 6 keyword datasets, each keyword containing 83 files. We divided each file in 198 windows. Moreover, we used 13 MFCC features. So, it created a 198*13*83*6 4D array. Now, a new audio recording was taken and MFCC was performed. And then, DTW was performed within the new recording and previous datasets. To compute similarities, a DTW cost is estimated, which is done by using DTW matrix calculation. The dataset which has the least DTW cost is the result.

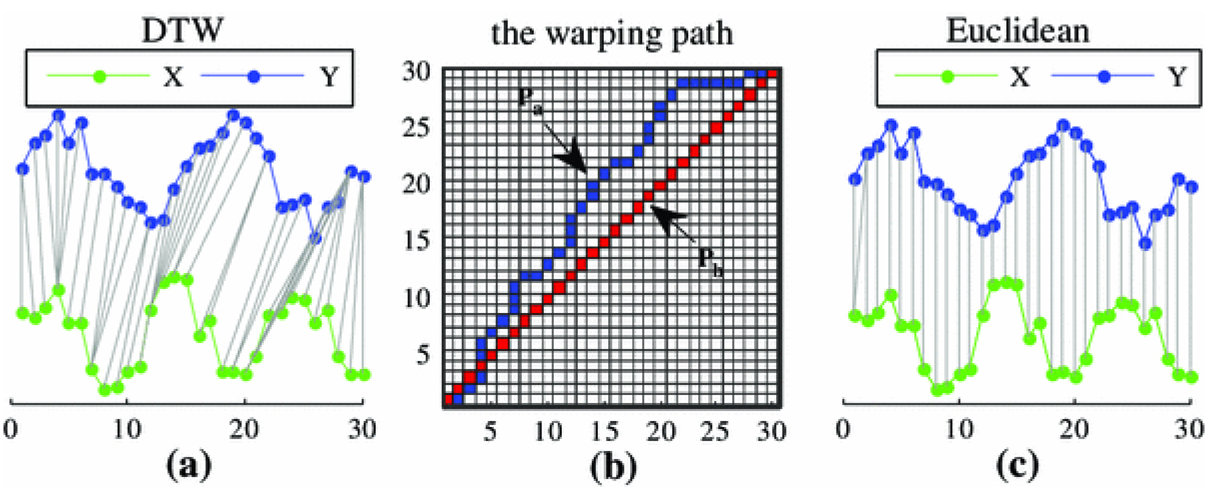


Figure 1: DTW visualization [2]

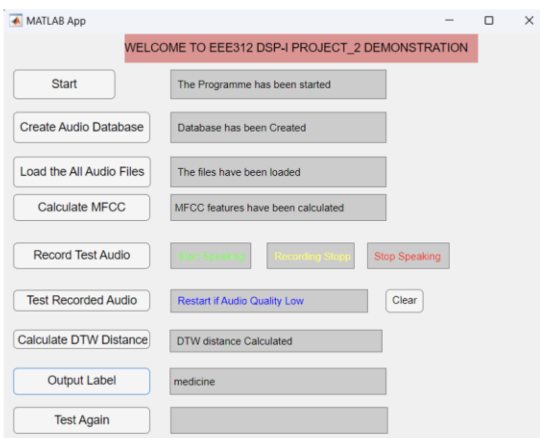
DTW was done using following code:

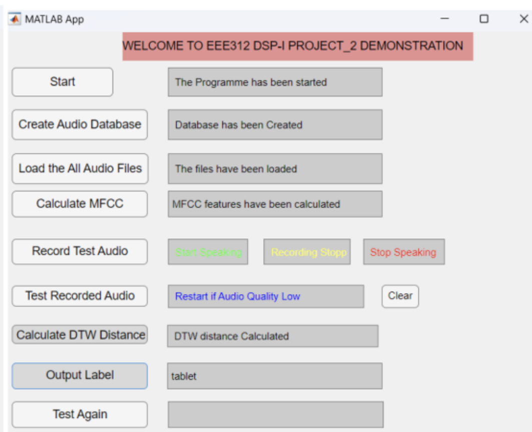
```
%% Compute DTW distances between the input audio and all the audio files
distances = zeros(1, length(labels));
for i = 1:length(labels)
    for j = 1:num_files

            distances(1,i) = distances(1,i) + dtw(M_input',
mfc_features(:,:,j,i)');

    end
end
```

Showing Result:

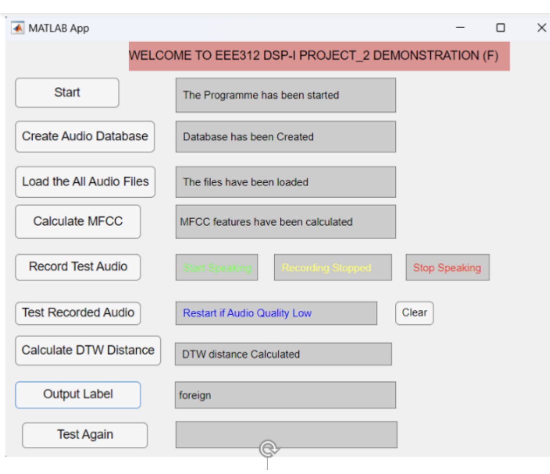
The label which has minimum DTW distance was shown as the result.

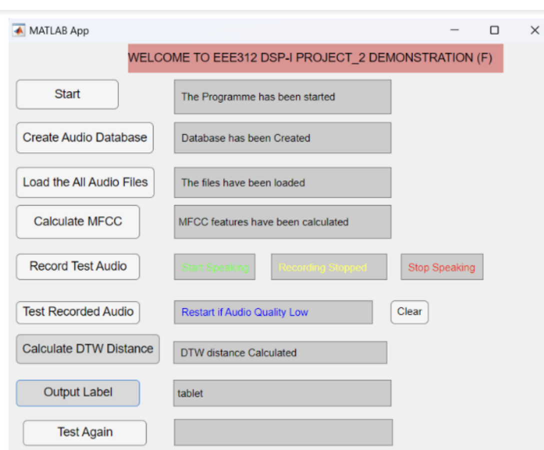
- Here some examples of Keyword detection without foreign word with App Designer





- Here some examples of Keyword detection with foreign word with App Designer





Here on the second one – PROJECT DEMONSTRATION (F) works with foreign keyword detection along with our selected local words.

Results

Our program detects spoken keywords almost precisely in ideal environment. However, in noisy environment, it shows some inaccuracy, especially while detecting these two keywords: “Fever”, “Syrup”.

The foreign keyword detection can label the foreign keywords as “Foreign”. However, sometimes it identifies the spoken keywords which are included in the dataset as foreign.

Here an accuracy table from several testing has been attached:

Test Type	Total test	Successful	Fail	% Success	%Fail
Without foreign Word	42	38	4	90.47	9.53
With Foreign Word	42	25	17	59.52	40.48

It noticeable that as a consequence of adding foreign data to our datasheet drastically reduced the accuracy. We have added several different words to Foreign label which dominates the other six labels.

Discussion

In real world, keyword detection from speech signal has application ranging from state-of-the-art surveillance to voice command-based services for the disabled people. The modern technology is evolving around taking the command or input from the speech. So, the project developed in this course is undoubtedly a time demanding area of future technological innovations using Digital Signal Processing.

However, throughout the undertaking of the project, it is noted that forming a perfect database is a crucial task which contributes significantly to the success rate of the keyword detection algorithm. On the other hand, profiling different types of noise based on the applied field and surroundings is also important to separate noise from the input signal.

The final challenge of such an application depending on uncomplicated algorithm is to work with different accents. As we have taken the keyword samples from different students in BUET whose birthplace were geographically miles apart; as a result, their accent was widely varied. It posed a great challenge to make the algorithm work with this database of voice samples having so many different accents.

Finally, the accuracy we have achieved is satisfactory and workable in small scale applications.

References

[1] https://link.springer.com/content/pdf/bbm:978-3-319-49220-9/1.pdf

[2] https://link.springer.com/article/10.1007/s13042-014-0254-0