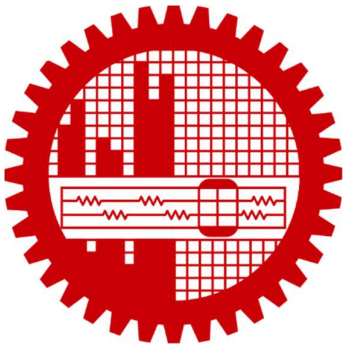


Bangladesh University of Engineering & Technology

KEYWORD RECOGNITION USING MATLAB

Course No: EEE 312

Course Title: Digital Signal Processing I Laboratory

Submitted to:

1. Shahed Ahmed
2. Barproda Halder

Submitted by:

1. Mir Noushad Hussain (1906090)
2. Irfan Ahmed (1906091)
3. Dipika Rani Nath (1906092)
4. Nafiz Imtiaz (1906093)
5. Md. Ashikur Rahman (1906094)

Section: B (B1)
Level: 3/ Term: 1
Group No: 07

Problem Statement

In this project we were given six key words. Those are book, marker, food, left, right, pencil. The user is going to say one of these words. Then we would have to detect which keyword the user said. The entire workflow has been described below.

PROPOSED METHODOLOGY

Data Collection

Firstly, we would have to collect data from as many people as possible. Most of the data are from hall residents. There are voices from both male and female students. The code for data collection is given below.

```
clc
clear all
close all

cd='D:\';
mkdir(cd,'Database')
cd=strcat(cd,'Database\');
id=input('identification(name or roll): ','s');
fprintf('\n');

record=['a';'b';'c';'d';'e';'f'];
str=["book" "marker" "food" "left" "right" "pencil" ];

p=1;
repeat='y';
while repeat~='n'

    for i=1:6
        fprintf('\n\npress any button to start recording: ')
        pause
        Fs=44100;
        nBits=16;
        nChannels=1;
```

```
        device=0;
        recObj = audiorecorder(Fs,nBits,nChannels,device);
        recTime=3;
        fprintf('recording started. now say %s\n' , str(i));
        recordblocking(recObj, recTime);
        fprintf('sound %d is recorded\n',i );
        myRecording = getaudiodata(recObj);
        cd1=strcat(cd,num2str(p),record(i),id,'.wav');
        audiowrite(cd1,myRecording,Fs);
    end

    fprintf('recording done %d times\n',p);
    repeat=input('Do another recording?(Enter n to quit, n==quit)==>
','s');
    p=p+1;
end
```

This program first creates a folder named Database. Within this it creates a sub folder with the name of the voice giver. This program usually takes 18 voice inputs per volunteer. Namely 6 keywords, 3 times each. The audio recorder function records the audio and stores it in a variable. The audio has been recorded in 44100 kHz. The audio is encoded in 16 bits. It's a single channel audio. Duration of each audio is 3 seconds. The audio files are recorded in .wav format. The audiowrite function saves the recorded voices in the desired folder. A folder filled with voice data typically looks like this.

Fig: Folder filled with voice data

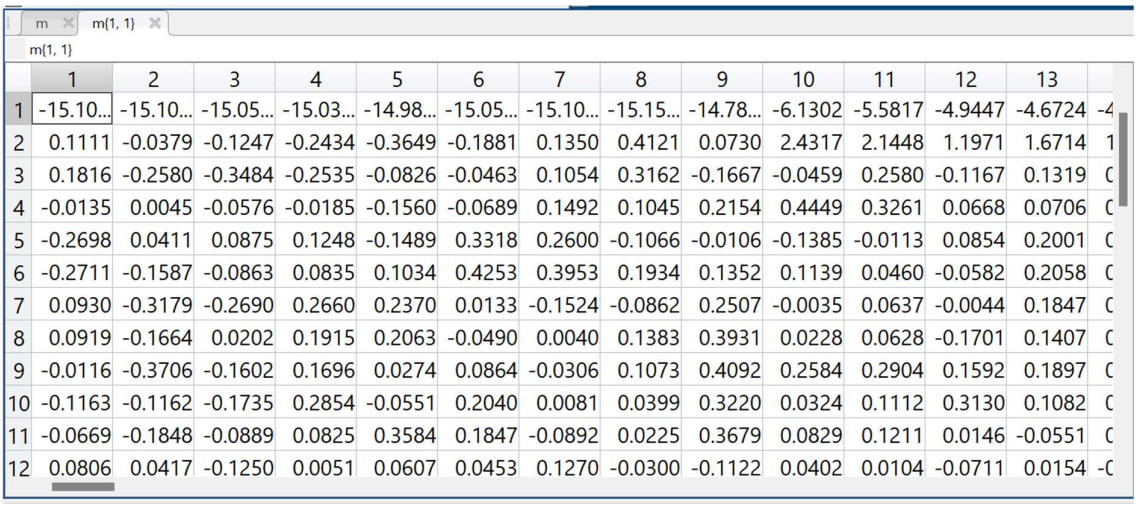
Building a Database

In this part of the project, we build a database from those audios. Directly storing all the audio signals would not be a good idea in this case. As they are going to

take a lot of space. This is going to make the program heavy on resources. Also, when comparing audio signals for final output, two audio signals cannot be compared directly. Because there is going to be lots of noise in the signals making the raw audio signals nearly unrecognizable. So we extract features using the mfcc function. It gives output the mel frequency cepstral coefficients. They are a total 13 in number. The first coefficient is replaced with log energy for this project. The code for feature extraction is given below.

```
clc;

close all;

% path
path_to_files = ['D:\Speech Recognition\Speech Recognition\drive-
download-20230225T081503Z-001\DTW_no_Loc\database\' ...
    '1906090'];

%  filenames
k = 1;
file_names = cell(18, 1);
for j = 1:3
    for i = 97:102
    file_names{k} = sprintf('%d%c1906090.wav', j, i);
    k = k+1;
    end
end

% read each file
for i = 1:length(file_names)
    % load audio file
    [audio, Fs] = audioread(fullfile(path_to_files, file_names{i}));
[coeffs,delta,deltaDelta]=mfcc(audio,Fs,LogEnergy="replace");
m{i+18*l} = [coeffs,delta,deltaDelta]';
end
l = l+1;
```

Along with mfcc, a total of 39 features were taken for this project namely delta and deltaDelta. A database of respectable size was built from this. A portion of the database is given below.



m | m{1, 1}

m{1, 1}

	1	2	3	4	5	6	7	8	9	10	11	12	13	
1	-15.10...	-15.10...	-15.05...	-15.03...	-14.98...	-15.05...	-15.10...	-15.15...	-14.78...	-6.1302	-5.5817	-4.9447	-4.6724	-4
2	0.1111	-0.0379	-0.1247	-0.2434	-0.3649	-0.1881	0.1350	0.4121	0.0730	2.4317	2.1448	1.1971	1.6714	1
3	0.1816	-0.2580	-0.3484	-0.2535	-0.0826	-0.0463	0.1054	0.3162	-0.1667	-0.0459	0.2580	-0.1167	0.1319	0
4	-0.0135	0.0045	-0.0576	-0.0185	-0.1560	-0.0689	0.1492	0.1045	0.2154	0.4449	0.3261	0.0668	0.0706	0
5	-0.2698	0.0411	0.0875	0.1248	-0.1489	0.3318	0.2600	-0.1066	-0.0106	-0.1385	-0.0113	0.0854	0.2001	0
6	-0.2711	-0.1587	-0.0863	0.0835	0.1034	0.4253	0.3953	0.1934	0.1352	0.1139	0.0460	-0.0582	0.2058	0
7	0.0930	-0.3179	-0.2690	0.2660	0.2370	0.0133	-0.1524	-0.0862	0.2507	-0.0035	0.0637	-0.0044	0.1847	0
8	0.0919	-0.1664	0.0202	0.1915	0.2063	-0.0490	0.0040	0.1383	0.3931	0.0228	0.0628	-0.1701	0.1407	0
9	-0.0116	-0.3706	-0.1602	0.1696	0.0274	0.0864	-0.0306	0.1073	0.4092	0.2584	0.2904	0.1592	0.1897	0
10	-0.1163	-0.1162	-0.1735	0.2854	-0.0551	0.2040	0.0081	0.0399	0.3220	0.0324	0.1112	0.3130	0.1082	0
11	-0.0669	-0.1848	-0.0889	0.0825	0.3584	0.1847	-0.0892	0.0225	0.3679	0.0829	0.1211	0.0146	-0.0551	0
12	0.0806	0.0417	-0.1250	0.0051	0.0607	0.0453	0.1270	-0.0300	-0.1122	0.0402	0.0104	-0.0711	0.0154	-0

The database was built using cell arrays. Each audio signal has a cell designated to itself. Each cell contains all 39 features of length 298.

Detection

Now comes the most important part of the project, detection. The code used for detection is given below.

```
clc
clear all
close all

format long;

% record new audio
Fs=44100;
nBits=16;
nChannels=1;
device=0;
recObj = audiorecorder(Fs,nBits,nChannels,device);
recTime=3;
fprintf('recording started\n');
recordblocking(recObj, recTime);
fprintf('sound is recorded\n');
```

```
audioIn = getaudiodata(recObj);

play(recObj)

[coeffs,delta,deltaDelta]=mfcc(audioIn,Fs,LogEnergy="replace");
newdata = [coeffs,delta,deltaDelta]';

% load feature matrix
load('db1.mat');

% find minimum distance
maxDist = Inf;
for i=1:length(m)
    dist = dtw(newdata,m{i});
    if dist < maxDist
        maxDist = dist;
        dist_data = i;
    end
end

% find detected word
word_level = rem(dist_data,6);
if word_level == 0
    word_level = 6;
end
words = {'book','marker','food','left','right','pencil'};
detected_word = words(word_level)
```

Firstly, we take the voice of the user who is testing this program. The user should say a particular keyword. After that we extract the 39 features and store that in a variable. Then we compare these features with all the cells in the database. We do the comparison using the dtw function. Its full form is dynamic time warping. It shows the minimum distance between the two signals to be compared. We find out which cell has the minimum distance from the newly received voice. The cell with the minimum distance is decided to be the cell with the features of the word that the test user has spoken. Then we find out which

keyword's features were on that cell using a small algorithm and print that word as the detected one.

Results

Output of a trial is given below.

```
detected_word =

  1×1 cell array

    {'pencil'}
```

The accuracy of the program was almost 60 percent. But it was not consistant.

Discussion

Keyword detection is an important part of digital signal processing. There are many applications of this project namely, attendance taking, security, vending machines etc. The accuracy of the project was not up to the mark. As not much data could be taken and due to the lack of knowledge of which features made those words distinguishable from one another. Machine learning can also be used for keyword detection. However, it requires a much larger database. This project can be further improved by fine tuning the algorithm. Especially experimenting with the features can give a much better accuracy. The word pairs book, food and left, right confused the algorithm a little as those words contain very similar parts. These word pairs require much more attention to features. Then it would be possible to detect all the words with a much better accuracy.