Indu Anuhya Rallabhandi (PES1201700795)

Maneesha S (PES1201700024)

Instructor's Name: P. Rama Devi

Course Title: Image processing using MATLAB

April 26, 2019

Speech recognition using MATLAB

Objectives:

The main purpose of our project is to check if a given audio file matches with an audio file present in our sample. We run a code using MATLAB which compares a given test file to all the audio files in the sample, to check whether it matched or not. This is useful in the real world for voice based passwords locks and similar applications. This is a very basic feature, whose complex versions are implemented in Apple's SIRI or Google's ALEXA.

Work done:

Our project, uses a code to check if a set of audio files match or mismatch. We have a set of sample files and a set of test files. When the code is run, a single test file is tested against the sample files and is checked if it matches. If it matches, it says match along with the test audio file. If it is a mismatch, it says mismatch. All the audio files are voice recordings from the people we know. We have asked a few people from our friends and family to record movie dialogues in their own voices, without any background noise. We have used all of the recordings for our sample set and selected our test set from the sample itself. We have 100 audio files in our sample set and about 15 audio files in our test set.

CODE:

```
unction speechrecognition(filename)
%Speech Recognition Using Correlation Method
%Write Following Command On Command Window
%speechrecognition(filename);
voice=audioread('test.wav');
x=voice;
x=x';
x=x(1,:);
x=x';
y1=audioread('one.wav');
y1=y1';
y1=y1(1,:);
y1=y1';
z1=xcorr(x,y1);
m1=max(z1);
l1=length(z1);
t1=-((l1-1)/2):1:((l1-1)/2);
t1=t1';
%subplot(3,2,1);
plot(t1,z1);
y2=audioread('two.wav');
y2=y2';
y2=y2(1,:);
y2=y2';
z2=xcorr(x,y2);
m2=max(z2);
l2=length(z2);
t2=-((l2-1)/2):1:((l2-1)/2);
t2=t2';
%subplot(3,2,2);
figure
plot(t2,z2);
y3=audioread('three.wav');
y3=y3';
y3=y3(1,:);
y3=y3';
z3=xcorr(x,y3);
m3=max(z3);
l3=length(z3);
t3=-((l3-1)/2):1:((l3-1)/2);
t3=t3';
%subplot(3,2,3);
figure
plot(t3,z3);
y4=audioread('four.wav');
y4=y4';
y4=y4(1,:);
y4=y4';
z4=xcorr(x,y4);
m4=max(z4);
l4=length(z4);
t4=-((l4-1)/2):1:((l4-1)/2);
t4=t4';
```

```
%subplot(3,2,4);
figure
plot(t4,z4);
y5=audioread('five.wav');
y5=y5';
y5=y5(1,:);
y5=y5';
z5=xcorr(x,y5);
m5=max(z5);
l5=length(z5);
t5=-((l5-1)/2):1:((l5-1)/2);
t5=t5';
%subplot(3,2,5);
figure
plot(t5,z5);


m6=300;
a=[m1 m2 m3 m4 m5 m6];
m=max(a);
h=audioread('allow.wav');
if m<=m1
   soundsc(audioread('one.wav'),50000)
      soundsc(h,50000)
elseif m<=m2
   soundsc(audioread('two.wav'),50000)
      soundsc(h,50000)
elseif m<=m3
   soundsc(audioread('three.wav'),50000)
      soundsc(h,50000)
elseif m<=m4
   soundsc(audioread('four.wav'),50000)
      soundsc(h,50000)
elseif m<=m5
   soundsc(audioread('five.wav'),50000)
      soundsc(h,50000)

else
  soundsc(audioread('denied.wav'),50000)

end
```

Results and discussions:

The present code works when the two audio files(the test file and the sample file), are exactly, the same. The files need to be of the exact same length with exact number of pauses at corresponding positions. Any difference in the two files(even in milliseconds), results in a mismatch. When the two files are exactly the same; in terms of speed, length

and content; the output is a match. The voices of the two audios can be different as long as the length, speed and content are exactly the same.

Future work:

The scope for improvement in our project is that it has too many constraints to match the two audios, which needs to be changed. Changing the code, to make it match the two audios only based on the words in the content irrespective of the speed,, voice, number of pauses, length etc. is part of our future work.