Open-Lab Mini Project Report
Speech to Text, using MATLAB

Title:

To implement Speech Recognition using MATLAB.

Team: Abhijeet Prem U4ECE15002
Ashwin Sanjay U4ECE15014
Akshay Bharadwaj U4ECE15034

Description:

We planned to design a system which recognised a particular Speech sample for what it was, using MATLAB. The recognition is done by showing the result in text format. Our plan of action included making a Data set containing the parameters, namely Zero Crossing Point, of various speech samples that were recorded from different individuals. This Data set makes up the Training Data Set. Testing is done by making use of another set of speech samples from the same set of individuals. KNN Search algorithm is used to test the system to recognise the sample Speech signals.

Our first challenge came in the form of preparing the Training Data set. We decided to record the voices of six individuals saying the code words: 'A', 'S', 'Start' and 'Stop'. Twelve samples for each word from each individual was recorded. We decided to use fifty samples from each word to make the Data set, and the remaining for testing. Out of the six persons, one was chosen for purely testing. Zero Crossing Point was found for each sample. It's Mean and Standard Deviation were calculated.

```
%loading audio file
[a,fs1]=audioread('A.wav');
%finding zero crossing
za=zerocrossing(a,fs1);
ma=mean(za); Mean of signal
me=sqrt(var(za)); Standard Deviation of signal
```

For the data set, the computed values for is loaded into the code in the form of a dataset. The dataset is converted into a cell array to access the contents.

```
ds = dataset('XLSFile','dataset.xlsx'); Accessing the excel sheet and creating a data set.
C = dataset2cell(ds); Converting data set into a Cell array.
```

The Data set is separated into two sets. One containing the actual data while the other, the names of the audio samples.

```
C=C(2:length(C),:); Cell Array containing the data values.
species=cell(C(:,1)); Cell Array containing the names of the audio samples.
```

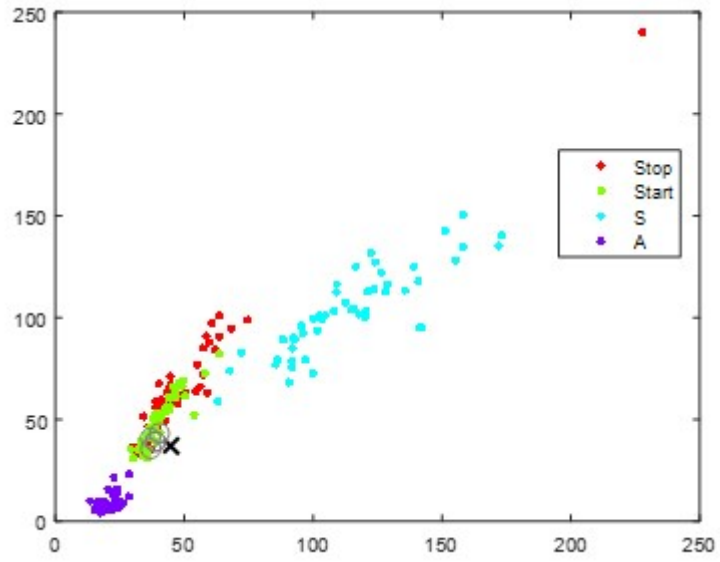
The creation of the data set makes up the training phase of our Objective. Our next step was to test the system for some sample signals. We made use of the KNN Search algorithm. The cell array containing the data values were converted to a Math array. The mean and Standard deviation of the Zero Crossing point of the sample signal made up the sample point. These parameters were put in the knnsearch function to figure out the sample.

```
x = cell2mat(C(:,2:3)); Converting cell array to math array
newpoint = [ma me]; Creating the Sample point using the parameters of ZCP
```

```
[n,d]=knnsearch(x,newpoint,'k',5);
```

In the above code, the sample point is compared against with all the values of the data set. The five closest values of the data set are considered and the sample can be regarded as the recorded Audio sample that has the maximum number of values among those five values. The following code is to show how the entire data is represented in a figure and how the required answer is obtained from the System.

```
gscatter(x(:,1),x(:,2),species); To create Scatter plot of the Dataset
line(newpoint(:,1),newpoint(:,2),'marker','x','color','k',...
'markersize',10,'linewidth',2)
line(x(n,1),x(n,2),'color',[.5 .5 .5],'marker','o',...
'linestyle','none','markersize',10);
set(legend,'location','best');
tb=tabulate(species(n,:)); This array contains Dataset that was picked out from the KNN
Search
m=cell2mat(tb(:,2));
[val,pos]=max(m);
tb(pos) To display the identity of the Audio sample that was tested
```



Other than this method of using an already recorded sample, we also made a code for recording an audio sample in real time for one second and test it. The following code shows how that was done.

```
% Record your voice for 1 second with Sampling Frequency 44100 Hz at 16 bits per sample.
recObj = audiorecorder(44100,16,1);
disp('Start speaking.')
recordblocking(recObj,1);
disp('End of Recording.');
play(recObj);
% Store data in double-precision array.
myRecording = getaudiodata(recObj);
```

Zero crossing Point was done using the following function.

```
Function[za] = zerocrossing(a,fs1)
%finding zero count of each segment
[A,n,t] = segment(a,fs1);
```

```
%generating hamming window
h=hamming(n);
za=zeros(t,1);


for i=1:t
    for j=1:n
      if j<n-1
          za(i) = za(i) + (abs(sign(A(i, j+1)) - sign(A(i,j)))) * h(j);
      end
    end
    za(i) = floor(za(i) / 2);
end
```

To Segment the input Audio signal

```
Function[x,n,t] = segment(amp,fs)
n=0.03 * fs;
off=floor(0.015 * fs);
v=1;
ln = length(amp);
t= floor(ln/(n-off));
%segmenting
for i=1:t
   for j=1:n
      if v<=ln
      x(i,j)=amp(v);
      v=v+1;
      else
       x(i,j)=0;
      end
   end
   v=v-off;
end
```

A portion of the Dataset is as follows:

Audio	Mean ZCP	SD ZCP
Stop	43.3396	63.3428
Stop	47.4340	57.6863
Start	49.1698	65.8655
Start	44.3019	53.9364
S	116.2051	103.9937
S	119.9487	100.2169
A	20.5385	15.5933
A	18.6923	6.2456

Testing:

Out of the twelve samples we recorded for each code word from each person two were used for testing

Audio Sample	File	Testing Result
A	Sample1	yes
	Sample2	yes
	Sample3	no
	Sample4	no
	Sample5	yes
	Sample6	yes
	Sample7	no
	Sample8	no
	Sample9	yes
	Sample10	yes
	Sample11	yes
	Sample12	yes

S	Sample13	yes
	Sample14	yes
	Sample15	yes
	Sample16	yes
	Sample17	yes
	Sample18	yes
	Sample19	yes
	Sample20	yes
	Sample21	yes
	Sample22	yes
	Sample23	yes
	Sample24	no

Start	Sample25	no
	Sample26	no
	Sample27	no
	Sample28	no
	Sample29	no
	Sample30	no
	Sample31	no
	Sample32	no
	Sample33	yes
	Sample34	yes
	Sample35	yes
	Sample36	yes

Stop	Sample37	yes
	Sample38	no
	Sample39	yes
	Sample40	yes
	Sample41	no
	Sample42	yes
	Sample43	yes
	Sample44	yes
	Sample45	no
	Sample46	yes
	Sample47	no
	Sample48	no

Accuracy:

No. of recognised samples = 30
No. of unrecognised samples = 18