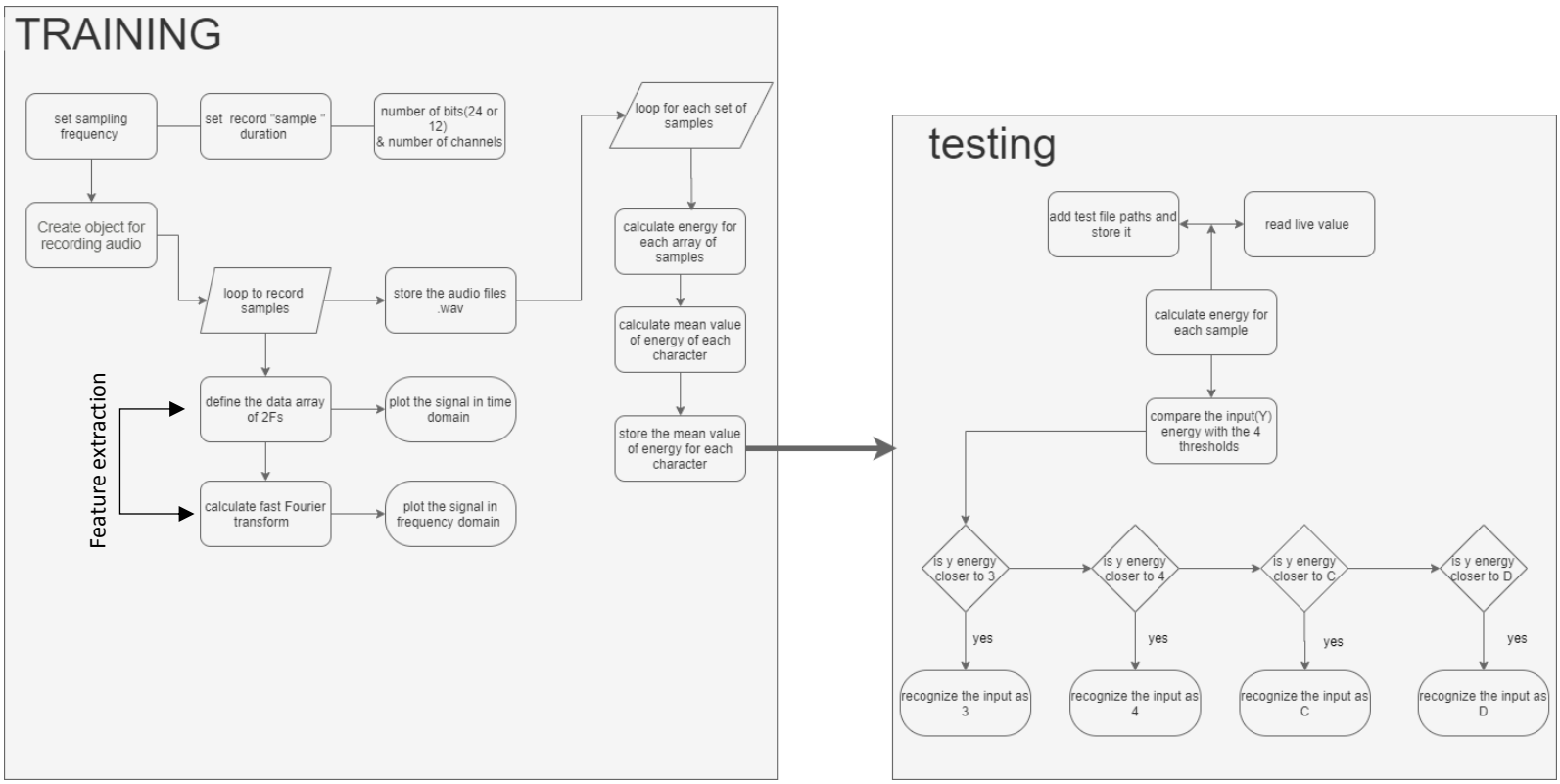
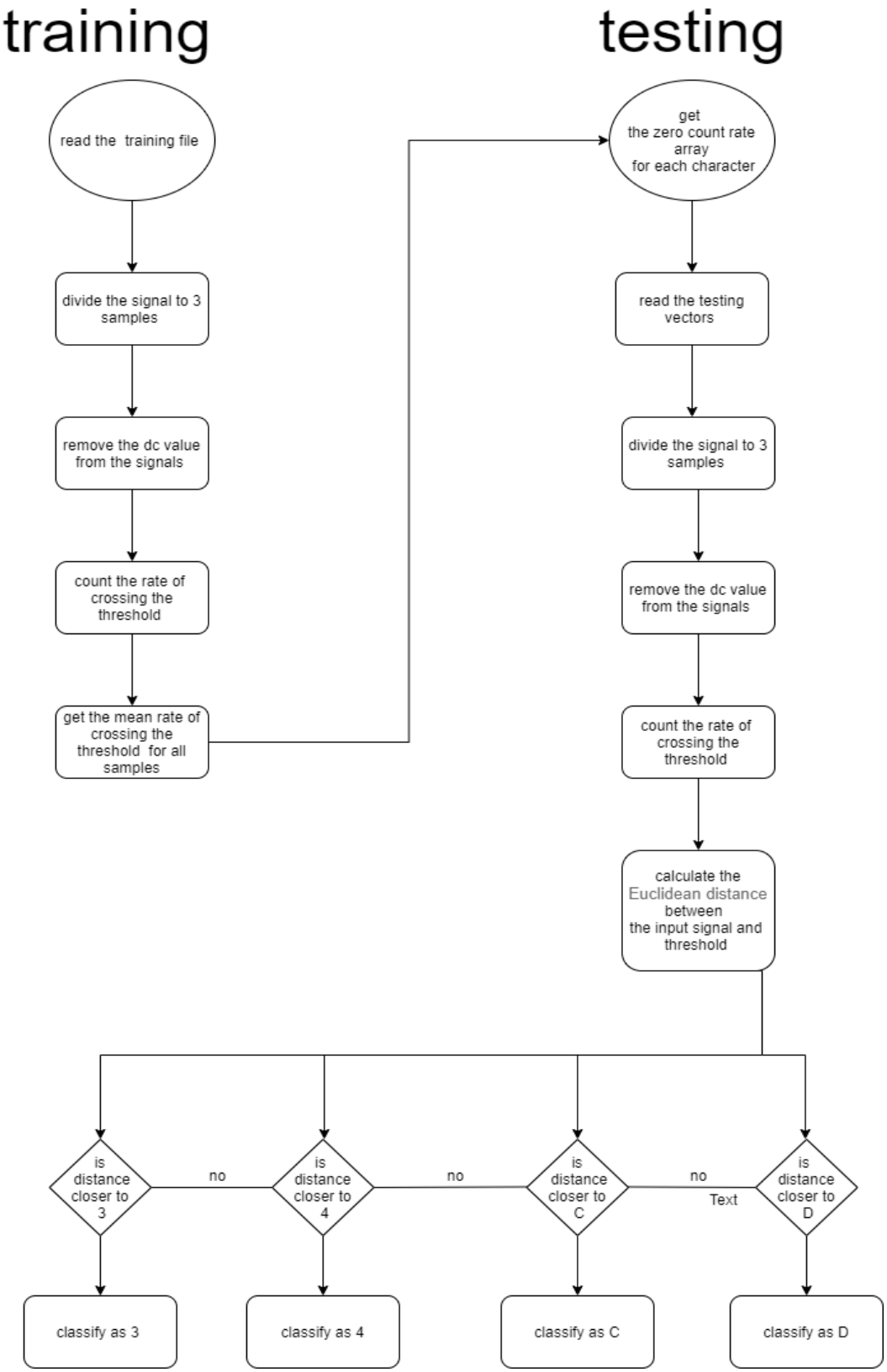
Part 2

method 1 block diagram:



method 2 block diagram :



solution explanation :

as the aim is to make my design covert voice into text and that means I have to set a specific output for specific input which means I have to recognize which input is that so I can give back output .

to recognize each signals of the characters I have to extract their features like frequency and energy
and to enhance the accuracy the extractions should be used on a set of signals

- first preprocessing is to record the samples from same source that will give the input from which is the same method of input “microphone “
- after recording 20 sample of each character of our classes [3 ,4 ,C,D] and storing it in separate paths
- get FFT is fast Fourier transform is an algorithm for efficient computation of discrete Fourier transform Which converts discrete periodic signal in time domain yd[n] to discrete periodic signal in frequency domain $X_n(k)$:

$$X_N(k) = \sum_{n=0}^{N-1} x_d[n] e^{\frac{-j2\pi kn}{N}}$$

N :number of samples

n: discrete time

k: discrete frequency

xd[n] :discrete time signal

Xn(k):discrete frequency signal

- then recorded the figures of signals in both time and frequency domains .
- calculate the energy of each signal $E = \sum^n (x_d)^2$.
- get the mean value of the energy for each class .
- store the value as threshold
- then for every incoming input we do fixed steps :
    - read the input
    - calculate the energy
    - Do comparison between each threshold of class [E_3 ,E_4 , E_C , E_D]
- Then determine the output character .

Method 2:

In that method we deal with the rate of zero crossing rate or the time access crossing rate :

- Applying sign function to the signal to signal y which returns 3 values :
    - 1 if the corresponding element of x is greater than 0 .
    - -1 if the corresponding element of x is less than 0 .
    - 0 if the corresponding element of x equals 0 .
- Use diff function calculates differences between adjacent elements of the signal array along the first array dimension whose size does not equal 1.
- Then get absolute value to remove the negative sign
- get the mean of them all to have the double count of the zero-access crossing rate so we have : x2 (cross/sample) rate .
- after extracting the featured array from every character we move to testing procedures and extract every test sample array and then do the Euclidean-distance for each reference and the test sample array .
- the shortest distance of them referred as the classification character .

In method 3 : it is a merge of first and second methods only

- we add energy to the array of parts samples
- because of the energy is so much bigger value and that would make energy parameter dominant so we will not use the Euclidean-distance
- instead we will use the cosine of the angle between each of them so the least value will be classified as the character .

The results of test **: method 1**

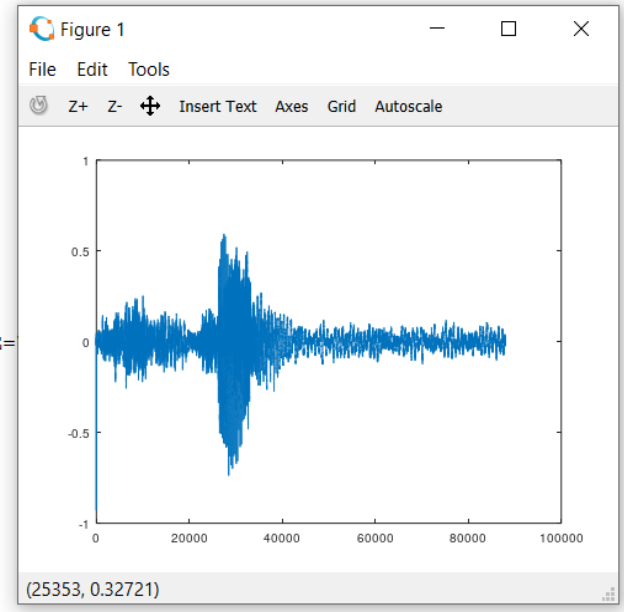
"input of 5 characters of each class : 3333 ,4444,CCCC,DDDD".

```
The energy of 3 is
 1989.7
The energy of 4 is
 2242.3
The energy of C is
 2167.2
The energy of D is
 2636.4
Test file [3] #1 classified as 3 E=1858.51
Test file [3] #2 classified as 3 E=1583.4
Test file [3] #3 classified as 3 E=1583.4
Test file [3] #4 classified as 3 E=1583.4
Test file [3] #5 classified as 3 E=1390.96
Test file [4] #1 classified as D E=5443.5
Test file [4] #2 classified as D E=2789.37
Test file [4] #3 classified as 4 E=2328.6
Test file [4] #4 classified as D E=2576.93
Test file [4] #5 classified as 4 E=2250.79
Test file [C] #1 classified as C E=2148.86
Test file [C] #2 classified as 4 E=2246.88
Test file [C] #3 classified as D E=2852.37
Test file [C] #4 classified as 4 E=2287.52
Test file [C] #5 classified as 3 E=1715.52
Test file [D] #1 classified as D E=2889.92
Test file [D] #2 classified as D E=2467.26
Test file [D] #3 classified as D E=3308.23
Test file [D] #4 classified as C E=2095.62
Test file [D] #5 classified as D E=2997.53
```

```
>> live_test

you will say same number 5 times

Start speaking for audio #1
Audio #1 has ended
Start speaking for audio #2
Audio #2 has ended
Start speaking for audio #3
Audio #3 has ended
Start speaking for audio #4
Audio #4 has ended
Start speaking for audio #5
Audio #5 has ended
Test file audio  classified as 3 E=
>>
```



The results of test **: method2**

```
The ZCR of C is
   0.054842   0.049396   0.025614
The ZCR of D is
   0.027974   0.022855   0.028134
The ZCR of 3 is
   0.033129   0.030250   0.029211
The ZCR of 4 is
   0.041047   0.026995   0.028119
Test file [3] #1 classified as 3
Test file [3] #2 classified as 4
Test file [3] #3 classified as 4
Test file [3] #4 classified as 4
Test file [3] #5 classified as C
Test file [4] #1 classified as D
Test file [4] #2 classified as 4
Test file [4] #3 classified as 4
Test file [4] #4 classified as 4
Test file [4] #5 classified as 4
Test file [C] #1 classified as C
Test file [C] #2 classified as C
Test file [C] #3 classified as C
Test file [C] #4 classified as C
Test file [C] #5 classified as C
Test file [D] #1 classified as D
Test file [D] #2 classified as D
Test file [D] #3 classified as D
Test file [D] #4 classified as D
Test file [D] #5 classified as D
>>
```

The results of test **: method 3**

```
>> zcrenergy

The ZCR of C is
      0.054842      0.049396      0.025614   2167.176474
The ZCR of D is
      0.027974      0.022855      0.028134   2636.366194
The ZCR of 3 is
      0.033129      0.030250      0.029211   1989.669409
The ZCR of 4 is
      0.041047      0.026995      0.028119   2242.333859
Test file [C] #1 classified as C
Test file [C] #2 classified as C
Test file [C] #3 classified as C
Test file [C] #4 classified as C
Test file [C] #5 classified as C
Test file [D] #1 classified as D
Test file [D] #2 classified as D
Test file [D] #3 classified as D
Test file [D] #4 classified as 4
Test file [D] #5 classified as D
Test file [3] #1 classified as 3
Test file [3] #2 classified as C
Test file [3] #3 classified as C
Test file [3] #4 classified as C
Test file [3] #5 classified as C
Test file [4] #1 classified as D
Test file [4] #2 classified as 4
Test file [4] #3 classified as 4
Test file [4] #4 classified as 4
Test file [4] #5 classified as 4
```

Analyzing method 1 results **using confusion matrix method :**

	Predicted 3	predicted 4	predicted C	predicted D	
Actual 3	**5**	**0**	**0**	**0**	**5**
Actual 4	**0**	**2**	**0**	**3**	**5**
Actual C	**1**	**2**	**1**	**1**	**5**
Actual D	**0**	**0**	**1**	**4**	**5**
	6	**4**	**2**	**8**	

Needs action

"3" recall = 5/5 =100% "4" recall = 2/5=40% "c" recall = 1/5=20% "d" recall = 4/5 =80%	"3" Precision =5/6 =83% "4" Precision =2/4=50% "C" Precision =1/2=50% "D" Precision = 4/8=50%
Model accuracy = (5+2+1+4)/ 20 =12/20=60%	

[actual 3 ,actual 4 ,actual c , actual d] are the actual value of the samples in test vector

[predicted "3,4,c,d"] are the predicted value of the given test vector

By using confusion matrix, we could the

- recall or "sensitivity " which means :
    - When it is actually the class , how often does it predict correctly?
- Precision:
    - When it predicts the class, how often is it correct?

From that analysis show up :

1. Class “character” 4 : have too much alias with character D that it predicts D more than 4
2. Class “character” C : does not have quite good threshold that band conflict with all characters that says why it have least recall percent of all .

After analyzing figured out we can get higher accuracy by :

- Increase the sample frequency F_s “ although that frequency was the max recommended F_s = 44100 Hz.
- Increase the training samples .
- Take group of same input together and get the mean as It was done in live-test
- Add another recognition metrics like the frequency features .

For data processing and visualizing the figures into ‘.png’ format small code was used .

Analyzing method 2 results **using <u>confusion matrix</u> :**

	Predicted 3	**predicted 4**	**predicted C**	**predicted D**	
Actual 3	**1**	**3**	**1**	**0**	**5** ← **Needs action**
Actual 4	**0**	**4**	**0**	**1**	**5**
Actual C	**0**	**0**	**5**	**0**	**5**
Actual D	**0**	**0**	**0**	**5**	**5**
	1	**7**	**6**	**5**	

“3” recall = 1/5 =20% “4” recall = 4/5=80% “c” recall = 5/5=100% “d” recall = 5/5 =100%	“3” Precision =1/1 =100% “4” Precision =4/7=57% “C” Precision =5/6=83% “D” Precision =5/5=100%
Model accuracy = (1+4+5+5)/ 20 =16/20=80%	

From that analysis show up :

3. Class “character” 4 : have too much alias with character 3 but fixed with D
4. Class “character” C : Precision and recall increased so much and gain accuracy of 100 % .
5. The character 3 that time totally lost .

We will try to merge both method 1 &2 to figure out if we will solve the “3” class problem .

Analyzing method 3 results **using confusion matrix :**

	Predicted 3	**predicted 4**	**predicted C**	**predicted D**	
Actual 3	**1**	**0**	**4**	**0**	**5** ← **Needs action**
Actual 4	**0**	**4**	**0**	**1**	**5**
Actual C	**0**	**0**	**5**	**0**	**5**
Actual D	**0**	**1**	**0**	**4**	**5**
	1	**5**	**9**	**5**	

“3” recall = 1/5 =20% “4” recall = 4/5=80% “c” recall = 5/5=100% “d” recall = 4/5 =80%	“3” Precision =1/1 =100% “4” Precision =4/5=80% “C” Precision =5/9=55% “D” Precision =4/5=80%
Model accuracy = (1+4+5+4)/ 20 =16/20=70%	

From that analysis show up :

1. Accuracy have not been improved but it decreased to 70% “better than method 1 and less than method 2 “
2. Class “character” D : Precision and recall affected and lost accuracy of 20 % .
3. The “character” 3 : stall have not improved .

These results are not constant it changes from testing samples to another and from character set to another .