Faculty of Engineering and Technology

Department of Electrical & Computer Engineering

ENCS4310: DIGITAL SIGNAL PROCESSING

Gender recognition and classification of speech signal

Prepared by: Zainab Jaradat,1201766

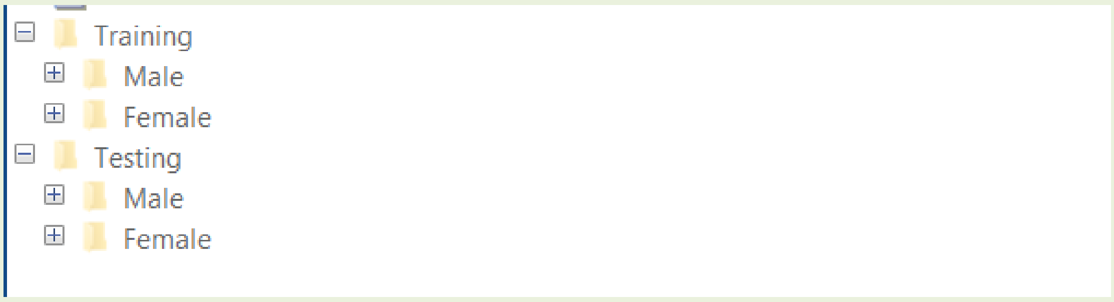
Instructor: Ashraf Al-Rimawi

Section: 1

Date: 24 January 2024

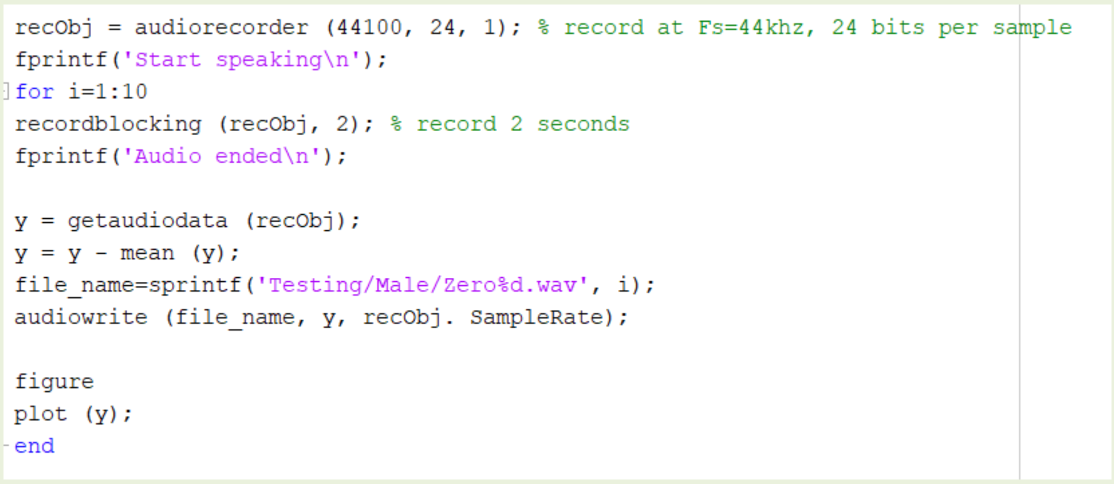
Code and Description:

First, I made two files : Training and Testing . I recorded more than 20 voice for each of male and female , and I tested 11 new file ".wav" for each of them.

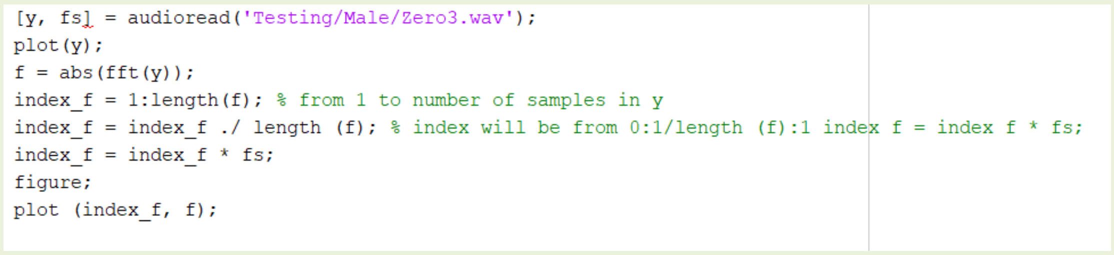


```
Training
    Male
    Female
Testing
    Male
    Female
```

I created this code to record 2 seconds of audio 10 times, saving each recording as a WAV file. I then visualize one of the recordings, showing its waveform and frequency spectrum. This helps me understand the characteristics of the recorded audio, making it useful for tasks like analyzing sound quality and frequency composition.



```
recObj = audiorecorder (44100, 24, 1); % record at Fs=44khz, 24 bits per sample
fprintf('Start speaking\n');
for i=1:10
recordblocking (recObj, 2); % record 2 seconds
fprintf('Audio ended\n');

y = getaudiodata (recObj);
y = y - mean (y);
file_name=sprintf('Testing/Male/Zero%d.wav', i);
audiowrite (file_name, y, recObj. SampleRate);

figure
plot (y);
end
```



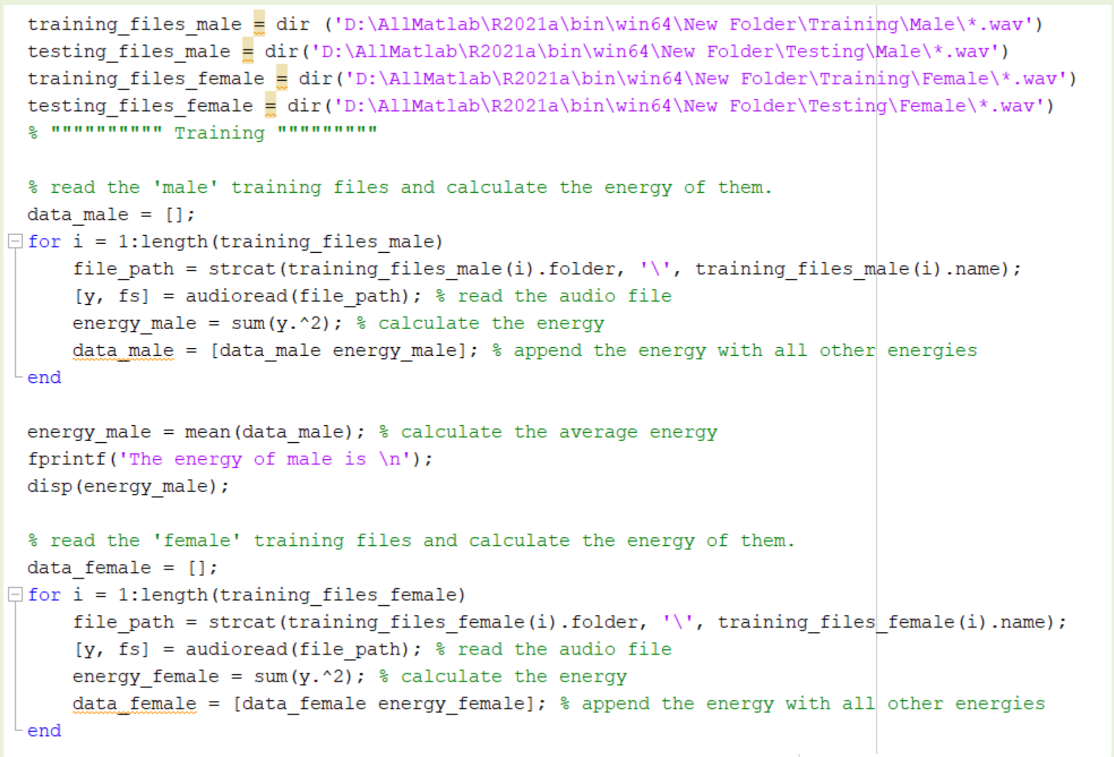
```
[y, fs] = audioread('Testing/Male/Zero3.wav');
plot(y);
f = abs(fft(y));
index_f = 1:length(f); % from 1 to number of samples in y
index_f = index_f ./ length (f); % index will be from 0:1/length (f):1 index f = index f * fs;
index_f = index_f * fs;
figure;
plot (index_f, f);
```

Then , I wrote this to classifying gender based on the energy of audio files. Here's what each part does:

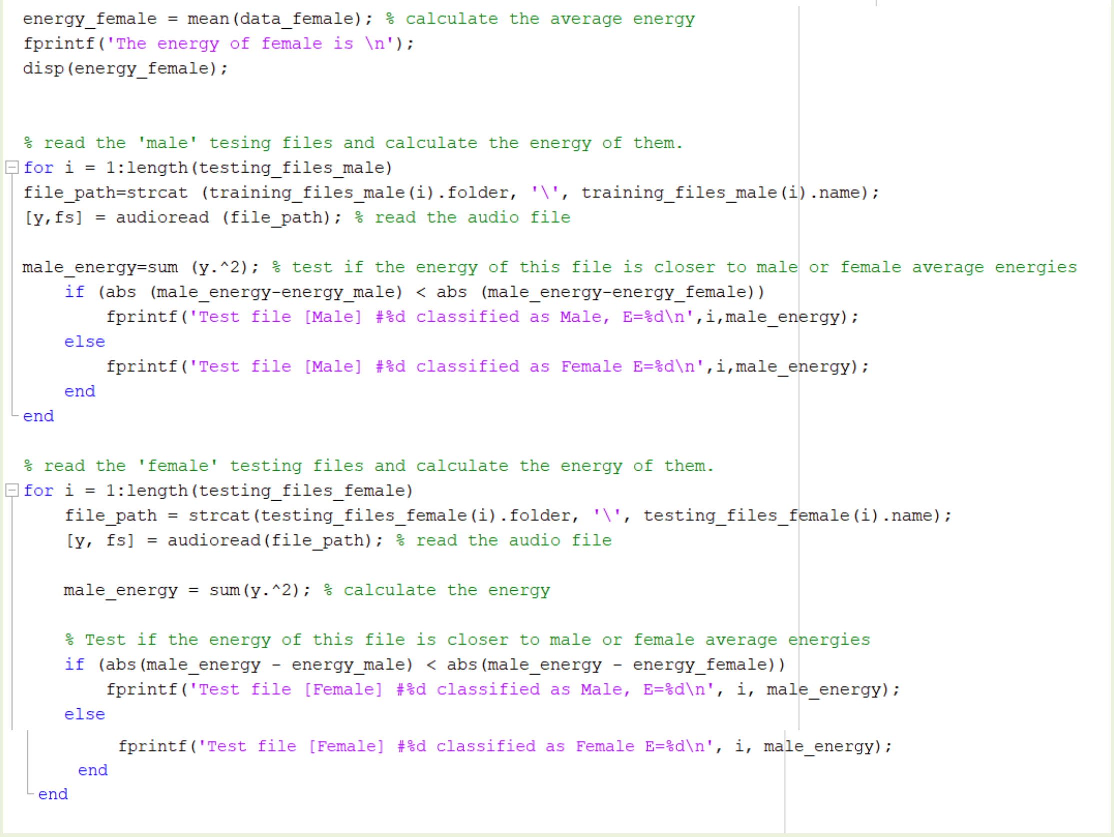
I gathered a list of male and female training and testing audio files. Then, I calculated the energy for each male and female training file and found the average energy for both genders. This helps me establish a baseline for male and female audio energy levels.

Next, I applied these average energy values to classify the gender of the testing files. For each male testing file, I calculated its energy and compared it to the average male and female energies. Based on the proximity, I classified the file as either male or female. The same process is repeated for female testing files.

The goal here is to explore whether the energy levels in audio files can be indicative of the gender of the speaker. Keep in mind that this is a basic approach, and more advanced methods might be explored for a more accurate classification.



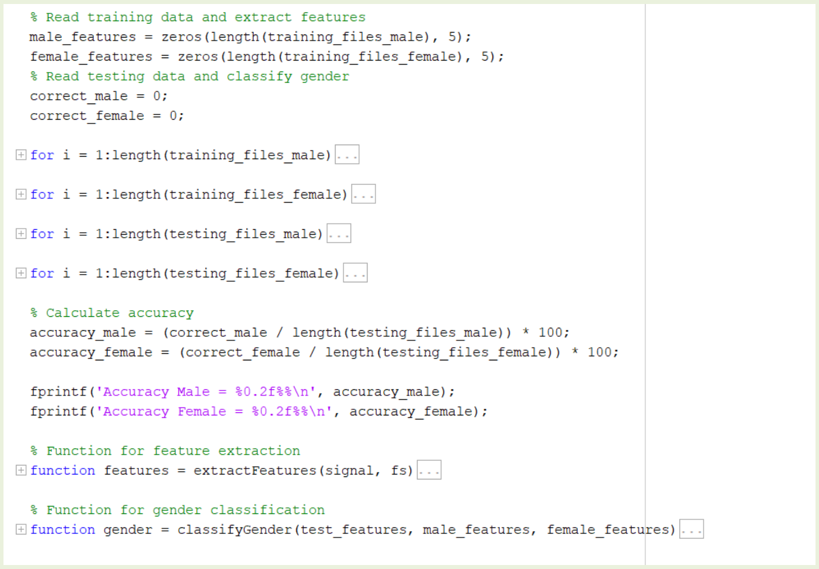
```
training_files_male = dir ('D:\AllMatlab\R2021a\bin\win64\New Folder\Training\Male\*.wav')
testing_files_male = dir('D:\AllMatlab\R2021a\bin\win64\New Folder\Testing\Male\*.wav')
training_files_female = dir('D:\AllMatlab\R2021a\bin\win64\New Folder\Training\Female\*.wav')
testing_files_female = dir('D:\AllMatlab\R2021a\bin\win64\New Folder\Testing\Female\*.wav')
% """"""""""" Training """"""""""

% read the 'male' training files and calculate the energy of them.
data_male = [];
for i = 1:length(training_files_male)
    file_path = strcat(training_files_male(i).folder, '\', training_files_male(i).name);
    [y, fs] = audioread(file_path); % read the audio file
    energy_male = sum(y.^2); % calculate the energy
    data_male = [data_male energy_male]; % append the energy with all other energies
end

energy_male = mean(data_male); % calculate the average energy
fprintf('The energy of male is \n');
disp(energy_male);

% read the 'female' training files and calculate the energy of them.
data_female = [];
for i = 1:length(training_files_female)
    file_path = strcat(training_files_female(i).folder, '\', training_files_female(i).name);
    [y, fs] = audioread(file_path); % read the audio file
    energy_female = sum(y.^2); % calculate the energy
    data_female = [data_female energy_female]; % append the energy with all other energies
end
```



```
energy_female = mean(data_female); % calculate the average energy
fprintf('The energy of female is \n');
disp(energy_female);


% read the 'male' tesing files and calculate the energy of them.
for i = 1:length(testing_files_male)
file_path=strcat (training_files_male(i).folder, '\', training_files_male(i).name);
[y,fs] = audioread (file_path); % read the audio file

male_energy=sum (y.^2); % test if the energy of this file is closer to male or female average energies
    if (abs (male_energy-energy_male) < abs (male_energy-energy_female))
        fprintf('Test file [Male] #%d classified as Male, E=%d\n',i,male_energy);
    else
        fprintf('Test file [Male] #%d classified as Female E=%d\n',i,male_energy);
    end
end

% read the 'female' testing files and calculate the energy of them.
for i = 1:length(testing_files_female)
    file_path = strcat(testing_files_female(i).folder, '\', testing_files_female(i).name);
    [y, fs] = audioread(file_path); % read the audio file

    male_energy = sum(y.^2); % calculate the energy

    % Test if the energy of this file is closer to male or female average energies
    if (abs(male_energy - energy_male) < abs(male_energy - energy_female))
        fprintf('Test file [Female] #%d classified as Male, E=%d\n', i, male_energy);
    else
         fprintf('Test file [Female] #%d classified as Female E=%d\n', i, male_energy);
     end
 end
```

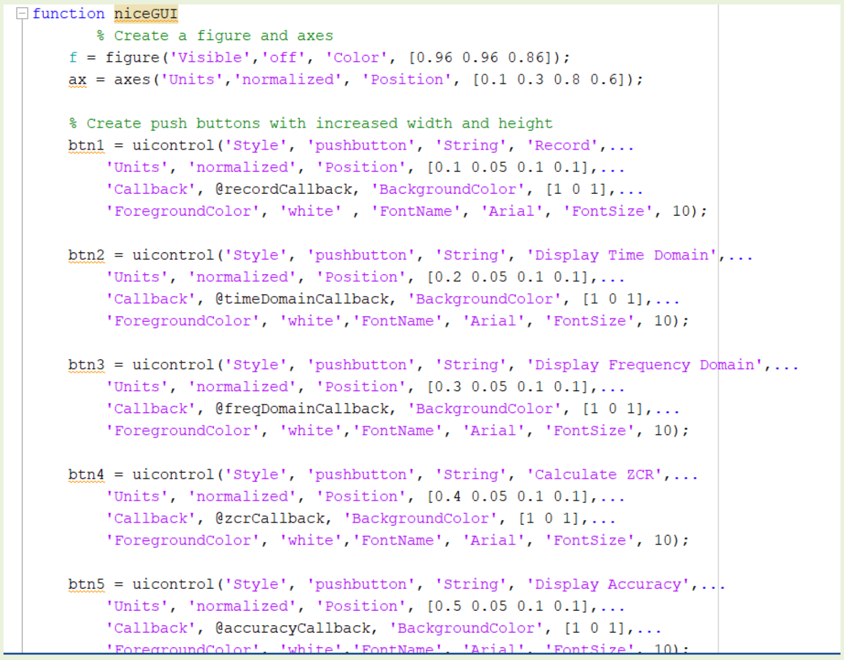
Then , I wrote code to classify gender from audio files with using a graphical user interface (GUI). In short, the code reads training and testing audio files, extracts features like Zero Crossing Rate (ZCR), energy, and Power Spectral Density (PSD) from each file. It then classifies the gender of testing files based on the features and displays the results, finally calculating and showing the accuracy of the classification for both male and female. The code aims to automate gender classification from audio data efficiently.

Code without GUI



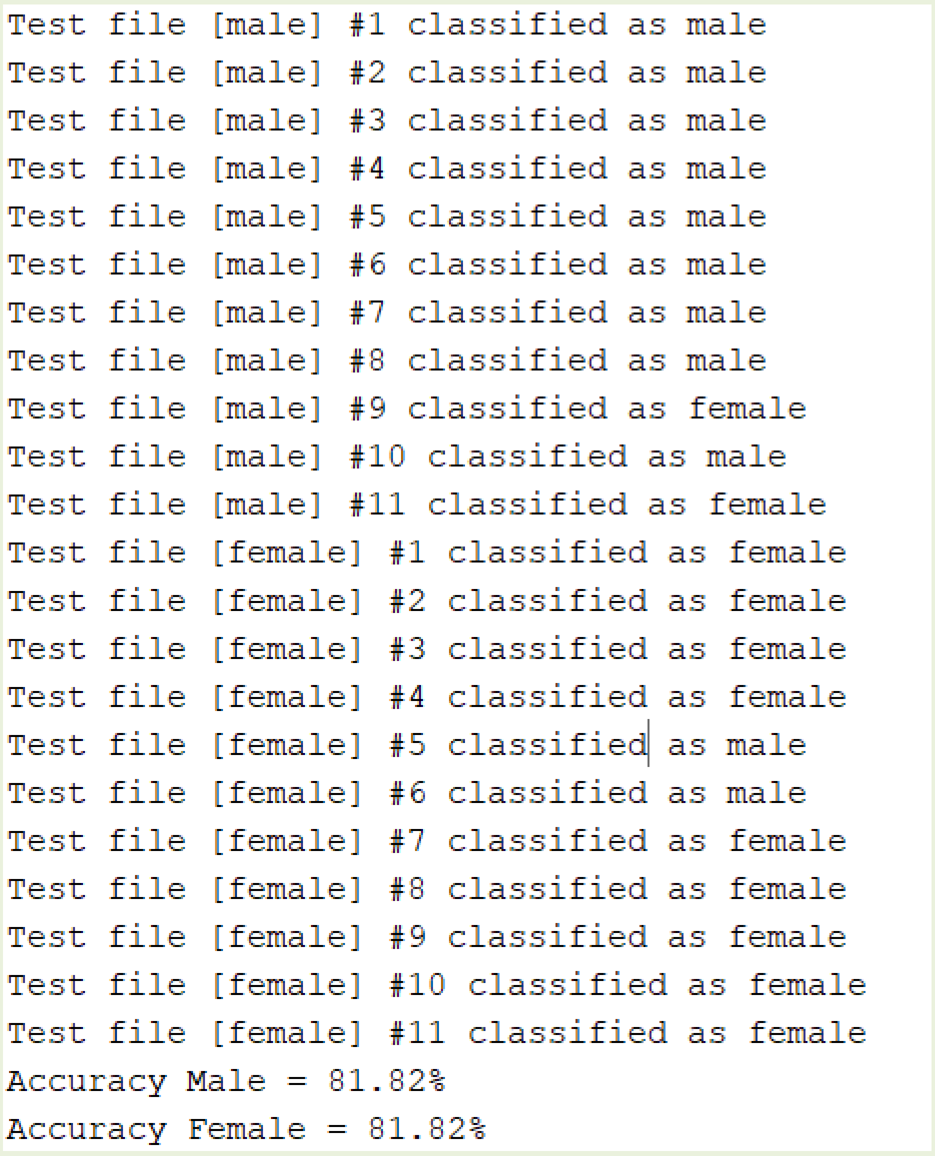
```
% Read training data and extract features
male_features = zeros(length(training_files_male), 5);
female_features = zeros(length(training_files_female), 5);
% Read testing data and classify gender
correct_male = 0;
correct_female = 0;

for i = 1:length(training_files_male)...

for i = 1:length(training_files_female)...

for i = 1:length(testing_files_male)...

for i = 1:length(testing_files_female)...

% Calculate accuracy
accuracy_male = (correct_male / length(testing_files_male)) * 100;
accuracy_female = (correct_female / length(testing_files_female)) * 100;

fprintf('Accuracy Male = %0.2f%%\n', accuracy_male);
fprintf('Accuracy Female = %0.2f%%\n', accuracy_female);

% Function for feature extraction
function features = extractFeatures(signal, fs)...

% Function for gender classification
function gender = classifyGender(test_features, male_features, female_features)...
```

Code with GUI



```
function niceGUI
        % Create a figure and axes
    f = figure('Visible','off', 'Color', [0.96 0.96 0.86]);
    ax = axes('Units','normalized', 'Position', [0.1 0.3 0.8 0.6]);

    % Create push buttons with increased width and height
    btn1 = uicontrol('Style', 'pushbutton', 'String', 'Record',...
        'Units', 'normalized', 'Position', [0.1 0.05 0.1 0.1],...
        'Callback', @recordCallback, 'BackgroundColor', [1 0 1],...
        'ForegroundColor', 'white' , 'FontName', 'Arial', 'FontSize', 10);

    btn2 = uicontrol('Style', 'pushbutton', 'String', 'Display Time Domain',...
        'Units', 'normalized', 'Position', [0.2 0.05 0.1 0.1],...
        'Callback', @timeDomainCallback, 'BackgroundColor', [1 0 1],...
        'ForegroundColor', 'white','FontName', 'Arial', 'FontSize', 10);

    btn3 = uicontrol('Style', 'pushbutton', 'String', 'Display Frequency Domain',...
        'Units', 'normalized', 'Position', [0.3 0.05 0.1 0.1],...
        'Callback', @freqDomainCallback, 'BackgroundColor', [1 0 1],...
        'ForegroundColor', 'white','FontName', 'Arial', 'FontSize', 10);

    btn4 = uicontrol('Style', 'pushbutton', 'String', 'Calculate ZCR',...
        'Units', 'normalized', 'Position', [0.4 0.05 0.1 0.1],...
        'Callback', @zcrCallback, 'BackgroundColor', [1 0 1],...
        'ForegroundColor', 'white','FontName', 'Arial', 'FontSize', 10);

    btn5 = uicontrol('Style', 'pushbutton', 'String', 'Display Accuracy',...
        'Units', 'normalized', 'Position', [0.5 0.05 0.1 0.1],...
        'Callback', @accuracyCallback, 'BackgroundColor', [1 0 1],...
        'ForegroundColor', 'white','FontName', 'Arial', 'FontSize', 10);
```

Results:



```
Test file [male] #1 classified as male
Test file [male] #2 classified as male
Test file [male] #3 classified as male
Test file [male] #4 classified as male
Test file [male] #5 classified as male
Test file [male] #6 classified as male
Test file [male] #7 classified as male
Test file [male] #8 classified as male
Test file [male] #9 classified as female
Test file [male] #10 classified as male
Test file [male] #11 classified as female
Test file [female] #1 classified as female
Test file [female] #2 classified as female
Test file [female] #3 classified as female
Test file [female] #4 classified as female
Test file [female] #5 classified as male
Test file [female] #6 classified as male
Test file [female] #7 classified as female
Test file [female] #8 classified as female
Test file [female] #9 classified as female
Test file [female] #10 classified as female
Test file [female] #11 classified as female
Accuracy Male = 81.82%
Accuracy Female = 81.82%
```

1. **For male testing files:**

   File #9 and #11 is misclassified as female. The rest are correctly classified as male

2. **For female testing files:**

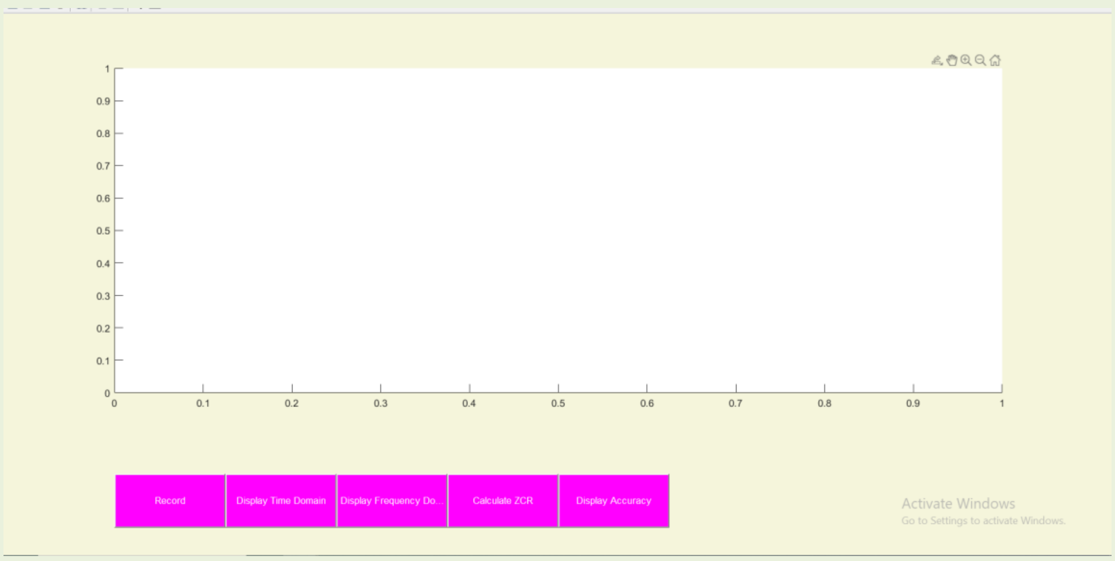
   File #5 and #6 is misclassified as female. The rest are correctly classified as male

3. **Overall Accuracy:**

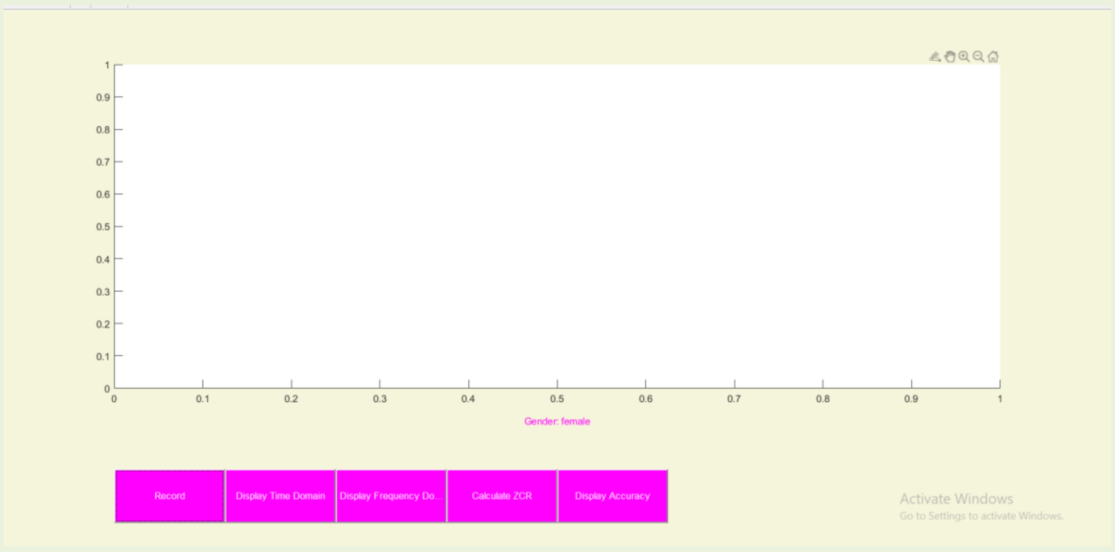
   - Accuracy for classifying male files is 81.82%.
   - Accuracy for classifying female files is 81.82%.

The classification is based on features such as Zero Crossing Rate (ZCR), energy, and Power Spectral Density (PSD). The code uses the cosine distance metric for classification, and the accuracy results provide insights into the effectiveness of the gender classification algorithm.

I implemented Matlab GUI application, which can do the recording, present the recorded signal in time domain and frequency domain, zero crossing account, and result accuracy.

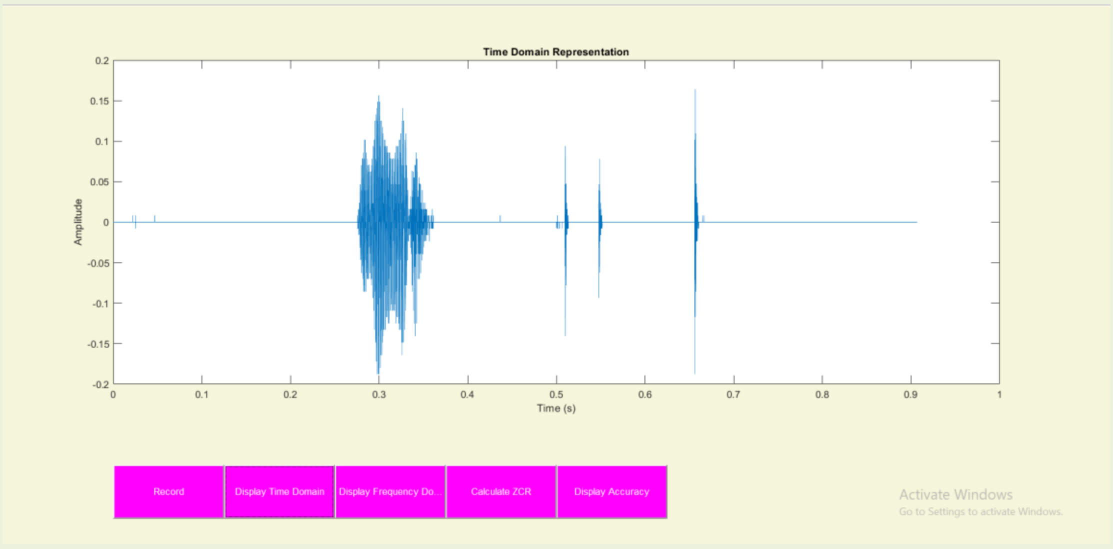


When I recored my voice , it prints "Gender : female"

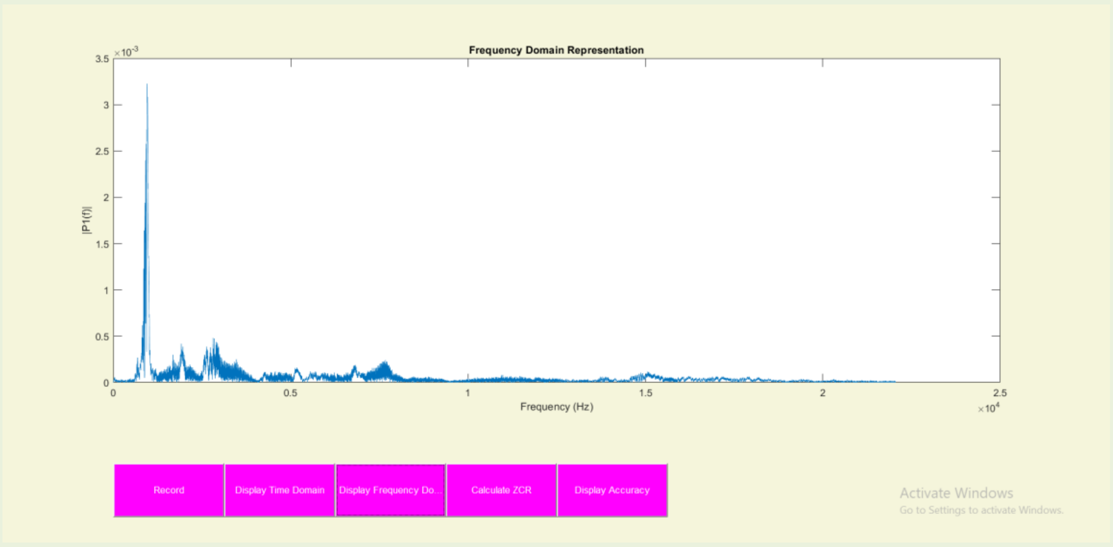


Gender: female

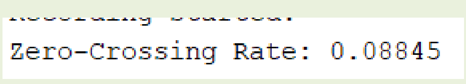
Time Domin:



Frequancy Domin:

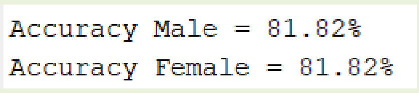


Zero crossing :



```
Zero-Crossing Rate: 0.08845
```

Accuracy:



```
Accuracy Male = 81.82%
Accuracy Female = 81.82%
```

Future works and :

I think the current system I’ve developed for gender classification based on voice characteristics is a good start. However, it could be improved and extended in several ways. By incorporating additional features such as Mel-Frequency Cepstral Coefficients (MFCCs), pitch, formant frequencies, spectral roll-off, and spectral flux, the system could capture a wider range of audio signal characteristics, potentially improving the accuracy of gender classification.

Furthermore, the use of machine learning models like Support Vector Machines (SVMs), Random Forests, or Neural Networks could provide better classification performance than a simple distance measure. Data augmentation techniques, such as adding noise or changing the pitch or speed of the audio signals, could also be used to enhance the robustness of the system.

Finally, the user interface could be improved to provide more feedback to the user, such as visualizing the extracted features or the classification process. In conclusion, these enhancements and extensions could lead to a more robust, accurate, and user-friendly system for gender classification based on voice characteristics.

Conclusion:

In conclusion, while the current system provides a good starting point for gender classification based on voice characteristics, there are many potential improvements and extensions that could enhance its performance and usability. By expanding the feature set, employing machine learning models, augmenting the training data, and enhancing the user interface, the system could be made more robust, accurate, and user-friendly.