BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

DSP (ENCS4310)

MATLAB Assignment

Gender Recognition System

Student's Name: Abdalkarim Eiss Student's Number: 1200015

Instructor: Dr. Ashraf Rimawi Section: 1

Date: 19/1/2024

Table of Contents

About The System

This system is identified the gender of speakers to male or female based on their audio recordings.

This system is identified the gender of speakers to male or female based on their audio recordings.

The system was trained on separate sets of audio files for males and females, extracting features such as Zero Crossing Rate, Energy, and Power Spectral Density to create a data. The system was built using MATLAB software tool.

Training Part

There are two training files, one for a male and the other for a female, each one has about twenty .wav file. These files were passed through a Butterworth filter (L. P. F) before it is be used. The idea of the training is to get the features of each gender to compare according it.

Testing Part

On this part, the system used fresh audio samples to apply the features it learned previously. The cosine distances between the test sample characteristics and the mean features of the training sets for male and female were calculated. The system assigned a gender to each test file based on the smallest cosine distance.

The Performance of the System

The system is displayed the results of each stage in addition to the accuracy of each gender classification. The female testing accuracy was 100.00%, also the male testing accuracy was 100.00%, this indicate that there is a correct classifying process. However, when the system was in its building kickstart, the energy was used as a feature only, and the accuracy was less than what it got now. So, as noted, as the features increased, the classification process will be more accurate.

Improvements

There were a few things that could be done better. It could be helpful to look into more speech-related topics, experiment with more complicated models, use a variety of training sets, modify parameters, use cross-validation techniques, and look into real-time processing capabilities in order to increase the system's accuracy and flexibility. Moreover, it could be helpful to use different words, this will increase the system classification accuracy.

Butterworth Filter Code

This filter was built using MATLAB tool to filter the recording audios which got from different people. Also, it is attached on a file.

```
%%Name: Abdalkarim Eiss --- ID: 1200015
%%%%This code is to filter the .wav voices sice it is
recorded using
%%%%mobiles
% Define the sampling frequency
fs = 44100;

% Design a low-pass filter (Butterworth filter)
order = 4; % Order of the filter
cutoff_frequency = 5000; % Cutoff frequency in Hz
[b, a] = butter(order, cutoff_frequency / (fs/2), 'low');

% Directory containing the .wav files
input_directory =
'C:/Users/Asus/Desktop/DSP_ass/OriginalAudio/Female';

% List of .wav files in the directory
file_list = dir(fullfile(input_directory, '*.wav'));

% Loop through each file and apply the filter
for i = 1:numel(file_list)
    % Load the audio file
    filename = fullfile(input_directory,
file_list(i).name);
    [audio, fs_original] = audioread(filename);
```

```
    % Ensure the audio is at the desired sampling frequency
    if fs_original ~= fs
        audio = resample(audio, fs, fs_original);
    end

    % Apply the filter
    filtered_audio = filter(b, a, audio);

    % Save the filtered audio to a new file
    output_filename =
fullfile('C:/Users/Asus/Desktop/DSP_ass/Train/FemaleF',
['filtered_', file_list(i).name]);
    audiowrite(output_filename, filtered_audio, fs);
end

disp('Filtering complete.');
```

The Gender Recognition System Code

Also, it is attached on a file. The results of this system are attached on a pdf file from the MATLAB command.

```
%%%Name: Abdalkarim Eiss
%%%ID: 1200015
%%%%Gender Recognition System
%%Training Definition
%%%Define the training male file directory
% Directory containing the .wav files
male_training_dir =
'C:\Users\Asus\Desktop\DSP_ass\Train\MaleF';
% List of male training .wav files in the directory
male_training_files = dir(fullfile(male_training_dir,
'*.wav'));
%%%Define the training female .wav directory files
female_training_dir =
'C:\Users\Asus\Desktop\DSP_ass\Train\FemaleF';
% List of male training .wav files in the directory
female_training_files = dir(fullfile(female_training_dir,
'*.wav'));
%%%%Testing definitions
%%%Define the testing male .wav directory files
```

```
male_testing_dir =
'C:\Users\Asus\Desktop\DSP_ass\Test\MaleF';
% List of male testing .wav files in the directory
male_testing_files = dir(fullfile(male_testing_dir,
'*.wav'));

%%Define the testing female .wav directory files
female_testing_dir =
'C:\Users\Asus\Desktop\DSP_ass\Test\FemaleF';
% List of male testing .wav files in the directory
female_testing_files = dir(fullfile(female_testing_dir,
'*.wav'));

%% ----------Training ------------
male_data = [];  % to store the features of male audio
training files
female_data = [];  % to store the features of female audio
training files

% Loop through each file for male training data
for i = 1:numel(male_training_files)
    % Load the audio file
    filename = fullfile(male_training_dir,
male_training_files(i).name);
    [y, fs] = audioread(filename);

    % Divide the signal into 3 parts and calculate features
for each part
    ZCR_m1 = mean(abs(diff(sign(y(1:floor(end/3)))))) ./2;
    ZCR_m2 = mean(abs(diff(sign(y(floor(end/3):floor
(end*2/3)))))) ./2;
    ZCR_m3 =
mean(abs(diff(sign(y(floor(end*2/3):end))))) ./2;
    energy_male = sum(y.^2); % Calculate energy
    % Power spectral density
    [psd, ~] = pwelch(y, [], [], [], fs);
    psd_male = mean(psd);

    % Combine features
    features_male = [ZCR_m1 ZCR_m2 ZCR_m3 energy_male
psd_male];
    male_data = vertcat(male_data, features_male(:));
end
```

```
% Calculate the mean of features for male training data
features_mean_male = mean(male_data);
fprintf('The features mean for Male Audios is \n');
disp(features_mean_male);

% Repeat the process for female training data
for i = 1:numel(female_training_files)
    % Load the audio file
    fName = fullfile(female_training_dir,
female_training_files(i).name);
    [y, fs] = audioread(fName);

    % Divide the signal into 3 parts and calculate features
for each part
    ZCR_f1 = mean(abs(diff(sign(y(1:floor(end/3)))))./2;
    ZCR_f2 = mean(abs(diff(sign(y(floor(end/3):floor
(end*2/3)))))./2;
    ZCR_f3 =
mean(abs(diff(sign(y(floor(end*2/3):end)))))./2;
    energy_female = sum(y.^2); % Calculate energy
    % Power spectral density
    [psd_f, ~] = pwelch(y, [], [], [], fs);
    psd_female = mean(psd_f);

    % Combine features
    features_female = [ZCR_f1 ZCR_f2 ZCR_f3 energy_female
psd_female];
    female_data = vertcat(female_data, features_female(:));
end

% Calculate the mean of features for female training data
features_mean_female = mean(female_data);
fprintf('The features mean for Female Audios is \n');
disp(features_mean_female);

%%%%%-------------Testing----------------
sum_m = 0;  % To count the successful male classifications
sum_f = 0;  % To count the successful female
classifications

fprintf('MALE Testing Results:\n');
% MALE TESTING
for i = 1:numel(male_testing_files)
    % Load the audio file
```

```
    fName = fullfile(male_testing_dir,
male_testing_files(i).name);
    [y, fs] = audioread(fName);

    % Divide the signal into 3 parts and calculate features
for each part
    ZCR_ma1 = mean(abs(diff(sign(y(1:floor(end/3)))))))./2;
    ZCR_ma2 = mean(abs(diff(sign(y(floor(end/3):floor
(end*2/3)))))))./2;
    ZCR_ma3 =
mean(abs(diff(sign(y(floor(end*2/3):end)))))./2;
    energy = sum(y.^2); % Calculate energy
    % Power spectral density
    [psd_y, ~] = pwelch(y, [], [], [], fs);
    psd_test = mean(psd_y);

    % Combine features
    features_test = [ZCR_ma1 ZCR_ma2 ZCR_ma3 energy
psd_test];

    % Calculate cosine distances
    cosine_dist_male = pdist2(features_test',
features_mean_male', 'cosine');
    cosine_dist_female = pdist2(features_test',
features_mean_female', 'cosine');

    % Make the decision based on cosine distance
    if (cosine_dist_male > cosine_dist_female)
        fprintf('Test file [Male] #%d classified as
FEMALE\n', i);
    else
        fprintf('Test file [Male] #%d classified as
MALE\n', i);
        sum_m = sum_m + 1;  % Calculate the sum of
successful male files
    end
end

%%% FEMALE TESTING
fprintf('FEMALE Testing Results:\n');
for i = 1:numel(female_testing_files)
    % Load the audio file
    fName = fullfile(female_testing_dir,
female_testing_files(i).name);
```

```
    [y, fs] = audioread(fName);

    % Divide the signal into 3 parts and calculate features
for each part
    ZCR_fe1 = mean(abs(diff(sign(y(1:floor(end/3)))))))./2;
    ZCR_fe2 = mean(abs(diff(sign(y(floor(end/3):floor
(end*2/3)))))))./2;
    ZCR_fe3 =
mean(abs(diff(sign(y(floor(end*2/3):end))))))./2;
    energy2 = sum(y.^2); % Calculate energy
    % Power spectral density
    [psd_y2, ~] = pwelch(y, [], [], [], fs);
    psd_test2 = mean(psd_y2);

    % Combine features
    features_test2 = [ZCR_fe1 ZCR_fe2 ZCR_fe3 energy2
psd_test2];

    % Calculate cosine distances
    cosine_dist_male2 = pdist2(features_test2',
features_mean_male', 'cosine');
    cosine_dist_female2 = pdist2(features_test2',
features_mean_female', 'cosine');

    % Make the decision based on cosine distance
    if (cosine_dist_male2 < cosine_dist_female2)
        fprintf('Test file [Female] #%d classified as
MALE\n', i);
    else
        fprintf('Test file [Female] #%d classified as
FEMALE\n', i);
        sum_f = sum_f + 1; % Calculate the sum of
successful female files
    end
end

%%%%Display and calculate the accuracy for each
classification process:
accuracy_m = (sum_m / length(male_testing_files)) * 100;
%%%To calculate the male classification accuracy
fprintf('Accuracy of classification for the male files is
%.2f', accuracy_m);
disp('%');
%%%Acuuracy calculations for the female classification
```

```
accuracy_f = (sum_f / length(female_testing_files)) * 100;
%%%To calculate the female classification accuracy
fprintf('Accuracy of classification for the female files is
%.2f', accuracy_f);
disp('%');
```