Doğa İlhan

CMPE468 Speech Processing

For this coding assignment, it was asked to analyze two speech files (should.wav and s5.vaw) and plot the measurements speech waveform, short-time energy, short-time magnitude, and short-time zero-crossing.

For these calculations, I implemented a MATLAB code (SpeechProcessing.m) and ran the code to get the asked calculations as output. Here are the screenshots for MATLAB code and its outputs:

```
windowSize = 1024;
windowShift = 512;

should = 'should.wav';
s5 = 's5.wav';

analyze(should, windowSize, windowShift);
analyze(s5, windowSize, windowShift);


function analyze(filePath, windowSize, windowShift)
    [speech, sampleRate] = audioread(filePath);

    energy = rms(buffer(speech, windowSize, windowSize - windowShift, 'nodelay'));

    magnitude = abs(spectrogram(speech, hann(windowSize), windowSize - windowShift));

    zeroCross = sum(abs(diff(sign(buffer(speech, windowSize, windowSize - windowShift, 'nodelay'))))) / windowSize;

    figure('Position', [100, 100, 800, 800]);

    subplot(4, 1, 1);
    plot(speech);
    xlabel('Sample');
    ylabel('Amplitude');
    title('Entire Speech Waveform');

    subplot(4, 1, 2);
    time = (windowSize/2 : windowShift : windowSize/2 + windowShift * (size(energy, 2)-1)) / sampleRate;
    plot(time, energy);
    xlabel('Time (s)');
    ylabel('Energy');
    title('Short-time Energy (En)');

```

Figure 1: Screenshot of MATLAB Code (1)

```
    subplot(4, 1, 3);
    time = (windowSize/2 : windowShift : windowSize/2 + windowShift * (size(magnitude, 2)-1)) / sampleRate;
    imagesc(time, [], 20*log10(magnitude));
    set(gca, 'YDir', 'normal');
    xlabel('Time (s)');
    ylabel('Frequency');
    title('Short-time Magnitude (Mn)');
    colorbar();

    subplot(4, 1, 4);
    time = (windowSize/2 : windowShift : windowSize/2 + windowShift * (length(zeroCross)-1)) / sampleRate;
    plot(time, zeroCross);
    xlabel('Time (s)');
    ylabel('Zero-crossing rate');
    title('Short-time Zero-crossing (Zn)');

    sgtitle(sprintf('Speech Analysis: %s', filePath));
end

```

Figure 2: Screenshot of MATLAB Code (2)

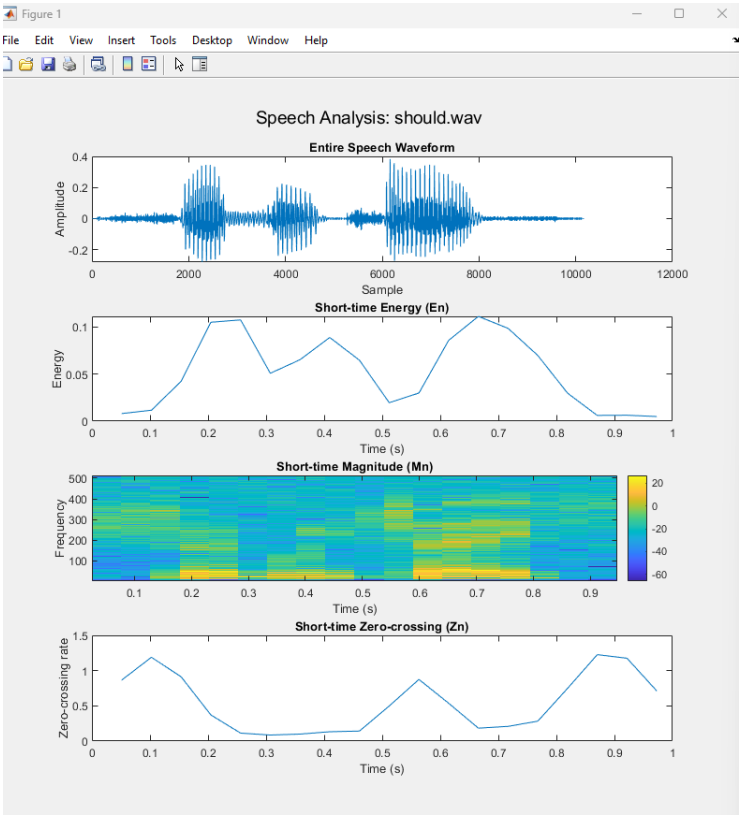

Figure 3: Calculations for should.wav

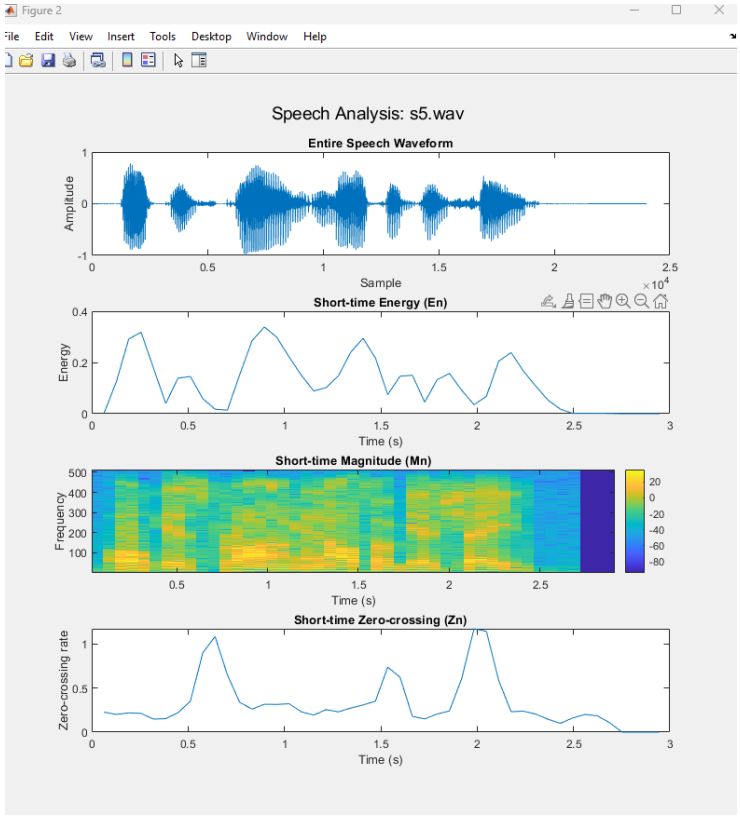

Figure 4: Calculations for s5.wav

The reasons behind chosen values of window sizes, window shifts, and window for the analysis can be explained as following:

1. Window Sizes: The window size defines how long each analytic window will be. The function uses a 1024-sample window size. The window size should be carefully selected to achieve a balance between temporal and frequency resolution. A wider window size provides better frequency resolution but worse time resolution, whereas a smaller window size provides better time resolution but lower frequency resolution. The 1024-sample size is a good compromise between time and frequency resolution for many speech analysis tasks.

2. Window Shifts: The window shift determines how many samples are shifted in successive analysis windows. The code uses a hop length of 512 samples. Choosing the proper hop length is essential for striking a balance between the amount of information acquired in the analysis and the processing performance. A shorter hop length offers finer temporal detail, but at the expense of additional windows and processing. While enhancing the temporal detail, a longer hop length reduces the cost of computing. Because it strikes a balance between maintaining computing efficiency and gathering enough temporal information, the 512 sample hop length was chosen.

3. Window: Each analysis window receives a window function to lessen spectral leakage and artifacts that may result from the window's abrupt start and end locations. A Hann window is used in the code (hann(window_size) used in calculation of magnitude, line 16 of the code). Due to its effective sidelobe attenuation and high frequency resolution, the Hann window is a common choice in speech analysis. It offers a reasonable middle ground between sidelobe suppression and main lobe breadth.