```
% TRAINING CODE
clear all
close all

load('F.mat'); % Load the dataset F
load('N.mat'); % Load the dataset N
load('O.mat'); % Load the dataset O
load('S.mat'); % Load the dataset S
load('Z.mat'); % Load the dataset Z

F = F(:,1:50); % Select part of the dataset F
N = N(:,1:50); % Select part of the dataset N
O = O(:,1:50); % Select part of the dataset O
S = S(:,1:50); % Select part of the dataset S
Z = Z(:,1:50); % Select part of the dataset Z

Fs = 173.61; % Defining the sampling frequency
N_shift = length(F); % Definig a variable to hold the length of the dataset
frequencies_shifted = (linspace(-pi*Fs, Fs*(pi - (2*pi)/N_shift), N_shift) + (Fs*pi)/(N_shift)*mod(N_shift, 2))'; % Compute the shifted frequencies for clarity in a

ffft = fft(F); % Computing the Fourier transform for the dataset F
ffft = fftshift(ffft); % Shifting the Fourier transform for clarity in analysis

nfft = fft(N); % Computing the Fourier transform for the dataset N
nfft = fftshift(nfft); % Shifting the Fourier transform for clarity in analysis

offt = fft(O); % Computing the Fourier transform for the dataset O
offt = fftshift(offt); % Shifting the Fourier transform for clarity in analysis

sfft = fft(S); % Computing the Fourier transform for the dataset S
sfft = fftshift(sfft); % Shifting the Fourier transform for clarity in analysis

zfft = fft(Z); % Computing the Fourier transform for the dataset Z
zfft = fftshift(zfft); % Shifting the Fourier transform for clarity in analysis


train = [sfft,ffft,nfft,offt,zfft]; % Creating a matrix of the fourier transformed datasets
[U,SS,VV] = svd(train,'econ'); % Find the eigenvectors of the train matrix
train_weights = U' * train; % Calculating the weights for each eigenvector to represent the train matrix

figure
% Use the Third EigenFFT for the 26 Hz signal
plot(frequencies_shifted,abs(U(:,3))) % Plot the eigenfrequencies
title('Eigenfrequency') % set the title of the plot
xlabel('Frequency [Hz]') % set the x-label of the plot
ylabel('Amplitude') % set the y-label of the plot
axis([-120 120 0 0.12]) % set the axis limits
```

TEST CODE

```
load('F.mat'); % Load the dataset F
load('N.mat'); % Load the dataset N
load('O.mat'); % Load the dataset O
load('S.mat'); % Load the dataset S
load('Z.mat'); % Load the dataset Z

F2 = F(:,51:end); % Select part of the dataset F
N2 = N(:,51:end); % Select part of the dataset N
O2 = O(:,51:end); % Select part of the dataset O
S2 = S(:,51:end); % Select part of the dataset S
Z2 = Z(:,51:end); % Select part of the dataset Z

ffft = fft(F2); % Computing the Fourier transform for the dataset F2
ffft = fftshift(ffft); % Shifting the Fourier transform for clarity in analysis

nfft = fft(N2); % Computing the Fourier transform for the dataset N2
nfft = fftshift(nfft); % Shifting the Fourier transform for clarity in analysis

offt = fft(O2); % Computing the Fourier transform for the dataset O2
offt = fftshift(offt); % Shifting the Fourier transform for clarity in analysis

sfft = fft(S2); % Computing the Fourier transform for the dataset S2
sfft = fftshift(sfft); % Shifting the Fourier transform for clarity in analysis

zfft = fft(Z2); % Computing the Fourier transform for the dataset Z2
zfft = fftshift(zfft); % Shifting the Fourier transform for clarity in analysis

test = [sfft,ffft,nfft,offt,zfft]; % Creating a matrix 'test' to hold all the fourier transformed datasets
test_weights = U' * test; % Find linear combinations of the eigenvectors to represent the test matrix

counter = 0; % Initializing the variable counter

for l=1:length(test_weights(1,:)) % Sweeping through the test_weights matrix
    % Using 51 eigen transforms we get 88.4% accuracy
    [dist,index] = min(vecnorm(test_weights(:,l) - train_weights(:,1:51))); % Computing the euclidian distance between the test and train sets
    if l <= 50 && (1 <= index) && (index <= 50) % Defining thresholds for accuracy calculation
```

```
        counter = counter + 1;
    elseif (51 <= l) && (51 <= index)
        counter = counter + 1;
    end
end
accuracy = counter / length(test_weights(:,1)) * 100 % Computing the accuracy and printing it in the Command Window
```

```
accuracy =

   88.4000
```

Published with MATLAB® R2019a