SEIZURE PREDICTION USING EEG

Our project analyzes EEG signals and classifies if there is a seizure present or not. In doing this, we hope to help people with epilepsy by aiding in the process of localizing a seizure origin.

CONTEXT AND ETHICS

According to CDC in 2015, about 3 million adults and 0.5 million children had active epilepsy. Intensive research is being carried out in Self-Management, Epidemiology, and Genomics of Epilepsy. General first-aid for a seizure includes timing the seizure, moving harmful objects away, put something soft under the patient's head, and to stay with the patient for the complete duration of the seizure.

Talking with some patients and researchers at the Mayo Clinic, we thought of making a system that could use the EEG data and process it to predict seizures by using historical EEG data. We wanted to train our algorithm to learn how to classify a piece of EEG data as seizure or non-seizure. This would be valuable as a part of a larger system that is capable of doing real-time processing and warning the doctors in the Epilepsy Monitoring Units at hospitals to make the process easier and more efficient.

In the Epilepsy Monitoring Unit, nurses and doctors will take patients off their medication to induce a seizure. Patients' real-time EEG readings are monitored by nurses and doctors who will inject a tracer into the patients with the onset of a seizure. They then will take a CT scan of the brain to find where blood is concentrated. During a seizure, there is more blood flow to the region of the seizure. By cross-referencing the CT scan with the EEG readings, doctors can localize the origin of the seizure.

However, a seizure will diffuse throughout the brain, and therefore, doctors must inject the tracer and take the CT scan quickly in order to have a more focused origin. We propose this system as a method to auto inject the tracer and trigger the CT scan in order to more quickly analyze the brain. This allows for a more

This technology would be implemented in the Epilepsy Monitoring Unit. We are proposing an autoinjection of the tracer and taking of the CT scan. When our technology detects a seizure within the EEG reading, the tracer would be autoadministered either through an IV or through a system similar to those used by diabetics. By automating the process, we hope to elimiate the time that can be critical to localizing the seizure without the need for nurses and doctors during the onset of a seizure.

Computational Tool

MATLAB

MATLAB® combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly. It includes the Live Editor for creating scripts that combine code, output, and formatted text in an executable notebook.

MAYO
CLINIC

Research

Mayo Clinic

Mayo Clinic conducts basic, translational, clinical and epidemiological research at its campuses in Arizona, Florida and Minnesota and throughout Mayo Clinic Health System.

Olin College of Engineering

EEG Research

Human Augmentation Lab

Our work aims to understand human attention and memory and to use this knowledge to develop personal technological devices that can respond to our intentions and cognitive states. Combining neuroscience and engineering, we conduct experiments about human behavior, record and analyze physiological signals, and explore emerging technologies.

AUTHORS

Sparsh Bansal

Thomas Jagielski

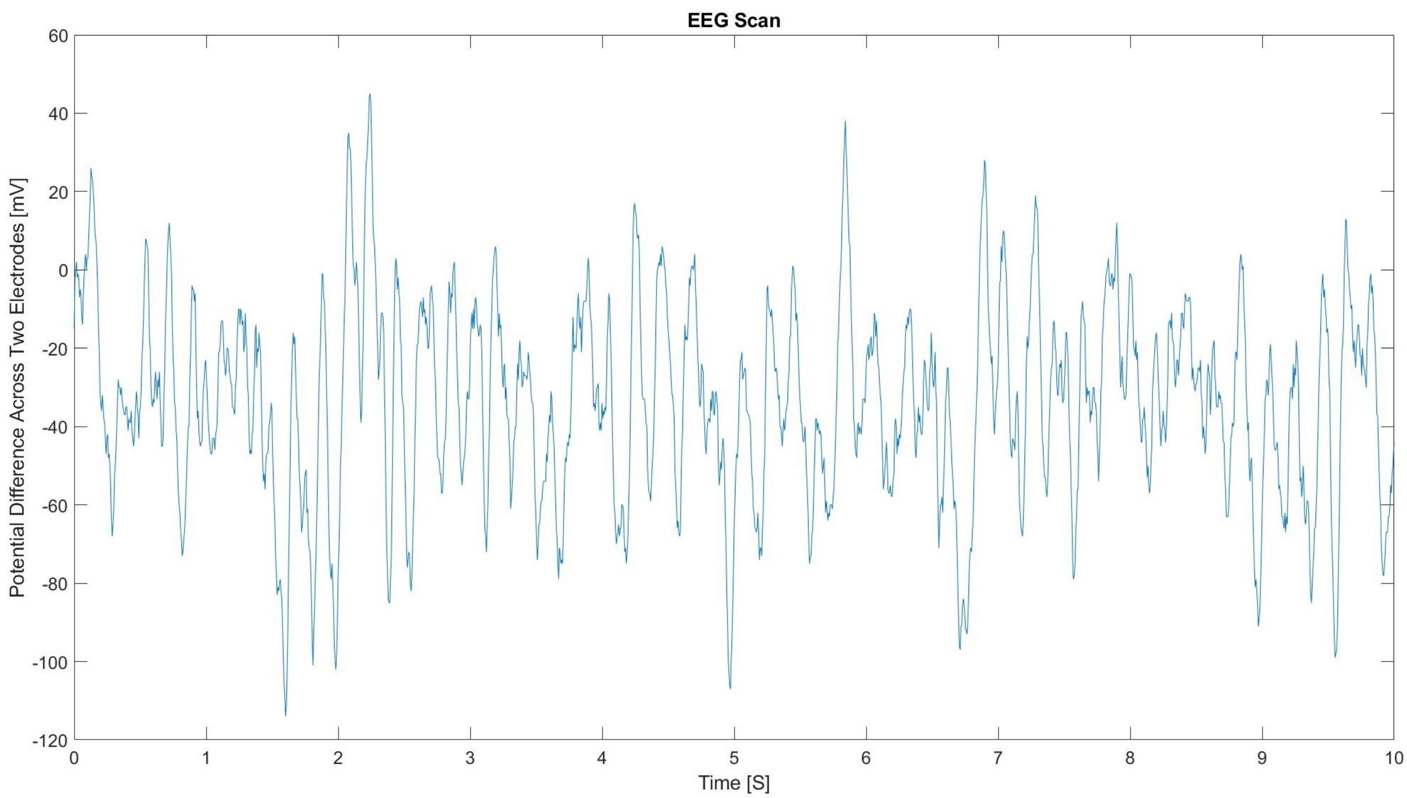
CONTACT

Suggestions? Potential Applications? Drop a note!

Olin College of Engineering, Needham MA
Email: sbansal, tjagielski (@olin.edu)

SCIENTIFIC BACKGROUND

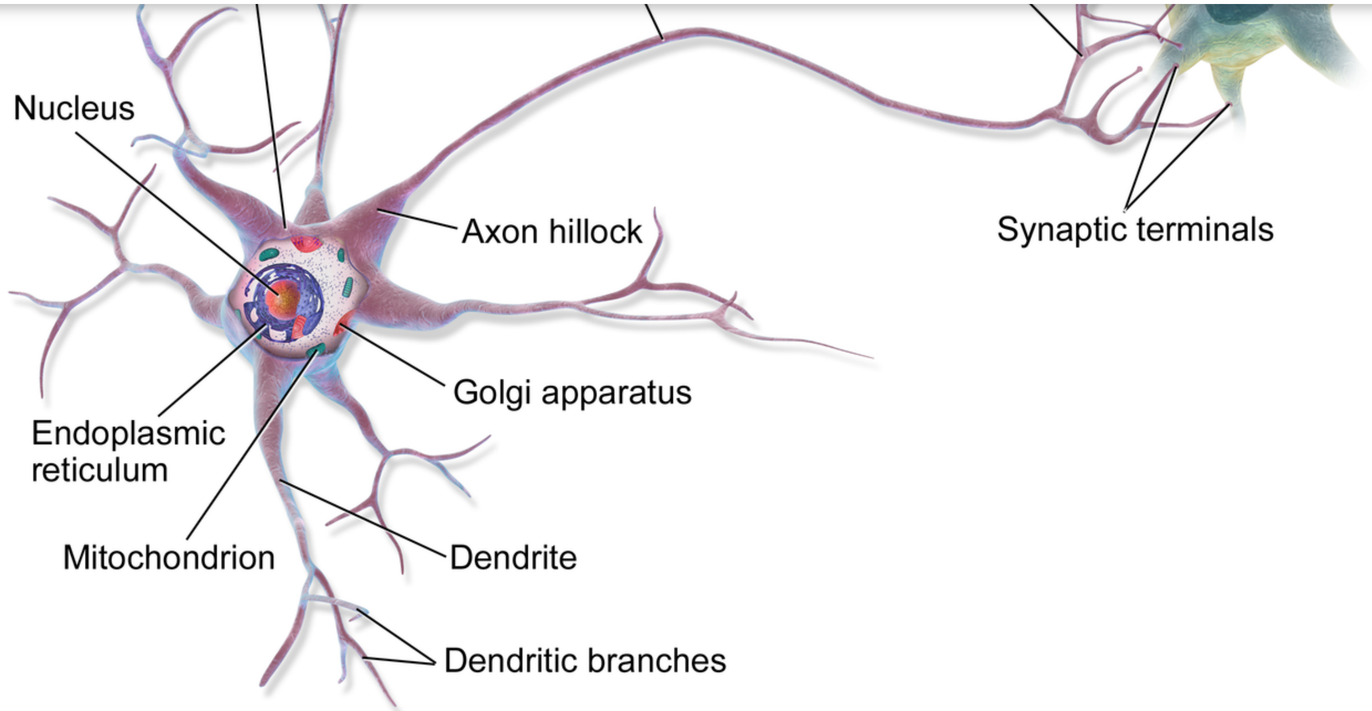
An EEG signal is a measurement of combined action potential waves between the measuring electrodes. An example of the EEG signal is shown below.



WHAT CAUSES AN EEG WAVE?

An action potential is the change in voltage overtime as a result of an electrical signal passing through a neuron. It does so in the following manner:

A. The charge in the cell is determined by Na+ ions.
B. The net charge reaches the threshold in the summation zone. In other words, the ions are summed at the Axon Hillock and, once triggered, the action potential travels down the axon.
C. Action potential reaches the sending end and the signal triggers the process in the next neurons.



The source of this image can be found at the following link: Image Source.

The resulting frequencies present in an EEG reading correspond with the delay between two action potentials. Therefore, higher frequencies can occur because more channels are opening quicker or the action potentials are firing at higher speeds. With many neurons firing, the same frequencies may occur across many neurons. This results in the frequencies present in the output EEG signal.

What Happens During a Seizure?

CONTACT

Suggestions? Potential Applications? Drop a note!

Olin College of Engineering, Needham MA
Email: sbansal@olin.edu

MATHEMATICAL BACKGROUND

For doing analysis on the EEG data, we use the eigenvector techniques for processing data, as well the fourier transforms for processing signals.

A. We compute the Shifted Fast Fourier Transform of the EEG datasets F(non-seizure), N(non-seizure), O(non-seizure), S(seizure), and Z(non-seizure). The fast fourier transform is computed by using a fast Fourier transform algorithm to compute the discrete fourier transform. The result of the above is then rearranged by MATLAB, which moves the zero-frequency component to the center of the array.

Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i \cdot 2\pi}{N} \cdot kn}$$

$$= \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi}{N} kn\right) - i \cdot \sin\left(\frac{2\pi}{N} kn\right)\right]$$

x_n is the signal

$\frac{2\pi \cdot i}{N}$ is each of the frequencies

X_k is the weight of each frequency

B. We create a list of frequencies that are shifted, or frequencies centered at zero radians per second. We do this so that the low frequency components are close to zero and the large frequency components are symmetric, and on the two edges.

C. We perform Singular Value Decomposition on the data in order to compress the data to weights to be used as linear combinations.

$M = U\Sigma V$

U is a matrix whose columns represent the eigenvectors of M^TM.

Σ is a diagonal matrix of the singular values (square-root of eigen values)

V is a matrix whose columns are the eigenvectors of MM^T

$$\omega = U^T M$$

U is the matrix that represents the weights for a linear combination in the eigen vector space.

D. We do some post-processing to normalize the data for easy visualizations.
E. We split the train and the test data to avoid biases in the accuracy calculation
F. We find the magnitude of the difference between the each of the test data and each of the train data sets to find out relations

$$\delta_k = \| \omega_{test_k} - \omega_{train_k} \|^2$$

δ_k is the Euclidean distance for a single sample.

ω_{test} is the weights of the test image set

ω_{train} is the weights of the train image set

G. We assign accuracy based on if the relations are matched correctly or not

CONTACT

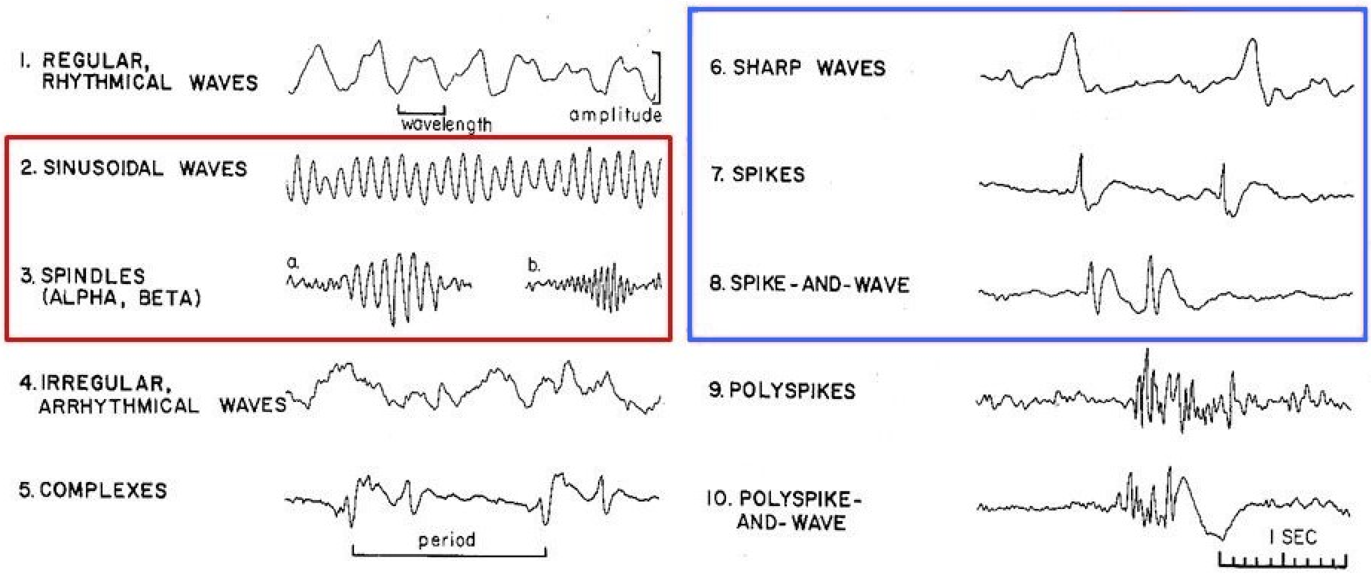
Suggestions? Potential Applications? Drop a note!

Olin College of Engineering, Needham MA
Email: sbansal, tjagielski (@olin.edu)

EEG MODEL

Common EEG Waveforms

Within EEG signals, there are 10 common waveforms you can expect to see. Three of these waveforms correspond with epileptic seizures. As indicated in the following image, the three main waves present in the signal are sharp waves, spike waves, and spike-and-wave. These, common in seizure, waveforms are boxed in blue. Common waveforms for normal brain activity are boxed in red.



This image is a modified version of one shown in a presentation by Dr. Valja Kellerovà. It can be found on slide 7 at the following link: Image Source.

Common EEG Frequencies

Normal brainwave frequencies can be classeified as:

A. Delta - frequency < 3 Hz, typically exist during a deep sleep

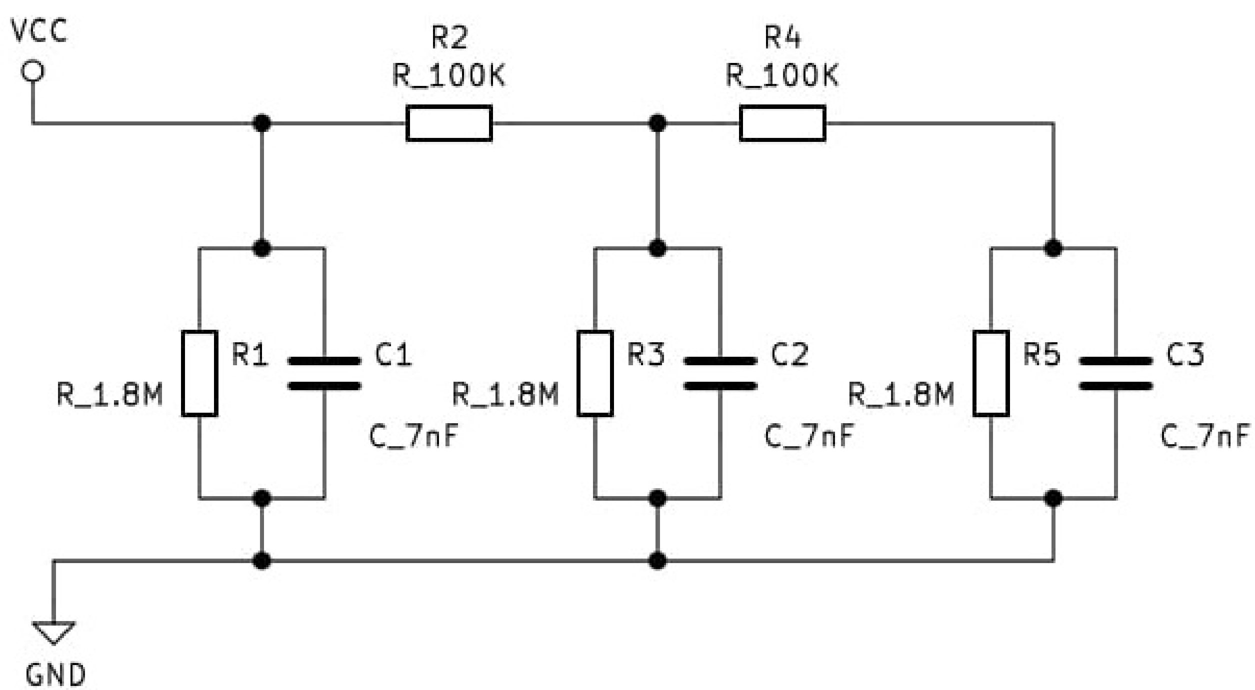
B. Theta - 3.5 Hz > frequency < 7.5 Hz, less deep sleep and uncommon for adults who are awake

C. Alpha - 7.5 Hz > frequency < 13 Hz, normal brain activity for an awake adult

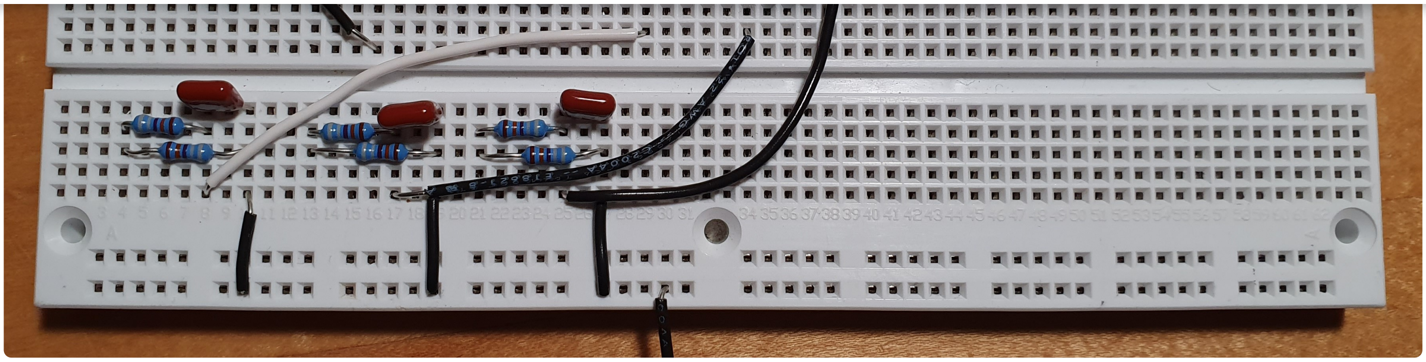
D. Beta - 14 Hz > frequency, typically found in adults who are anxious or alert

RC Circuit Model of a Neuron

Additionally, we can model a single neuron with resistors and capacitors. What this will model is an action potential curve. Since we know that the EEG signal is a combination of many action potentials, we could create a network with these RC circuits in them to model several neurons and the resulting signal. The following image shows the circuit schematic that we used to model a passive dendrite to observe the properties of a neuronal process.

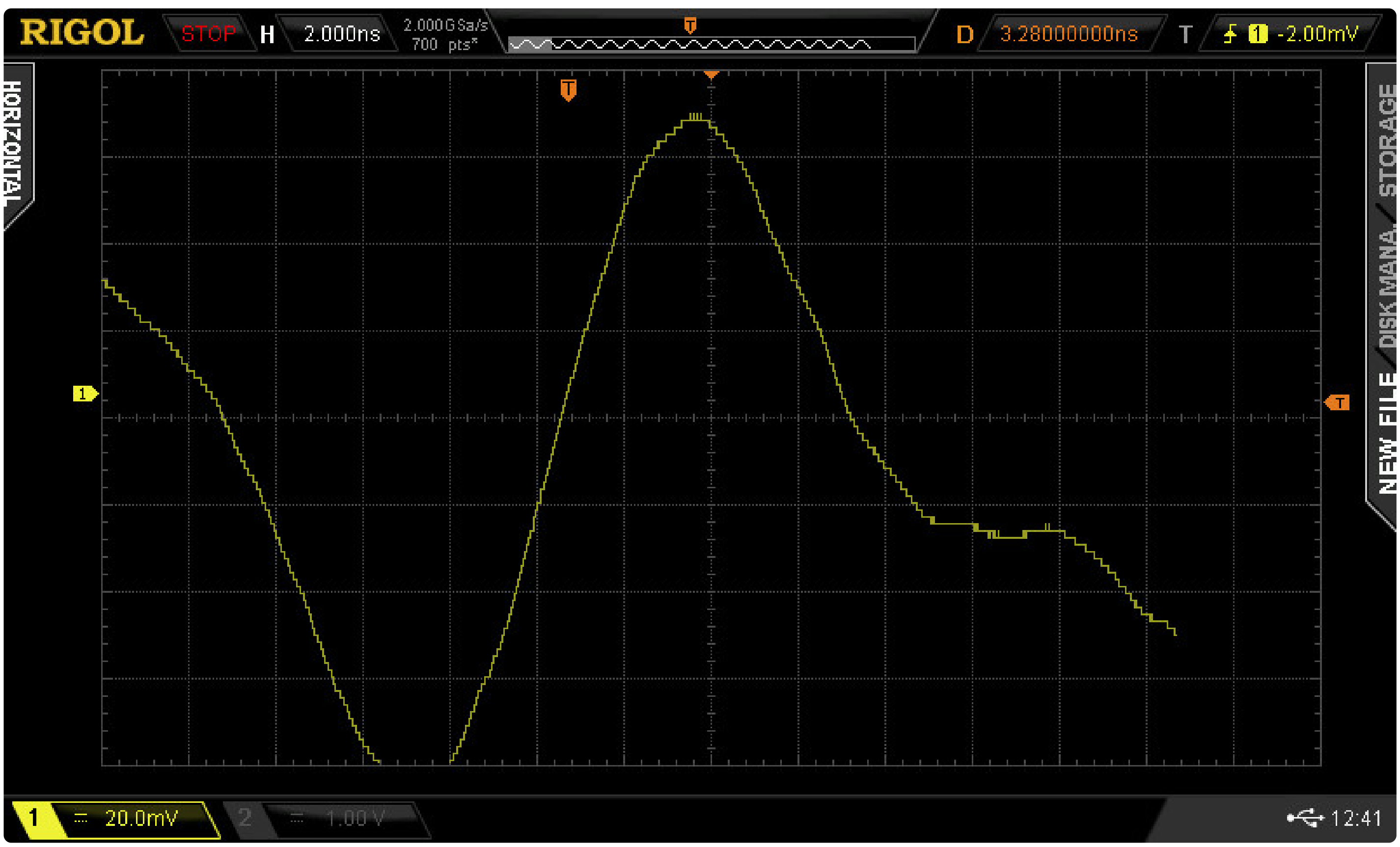


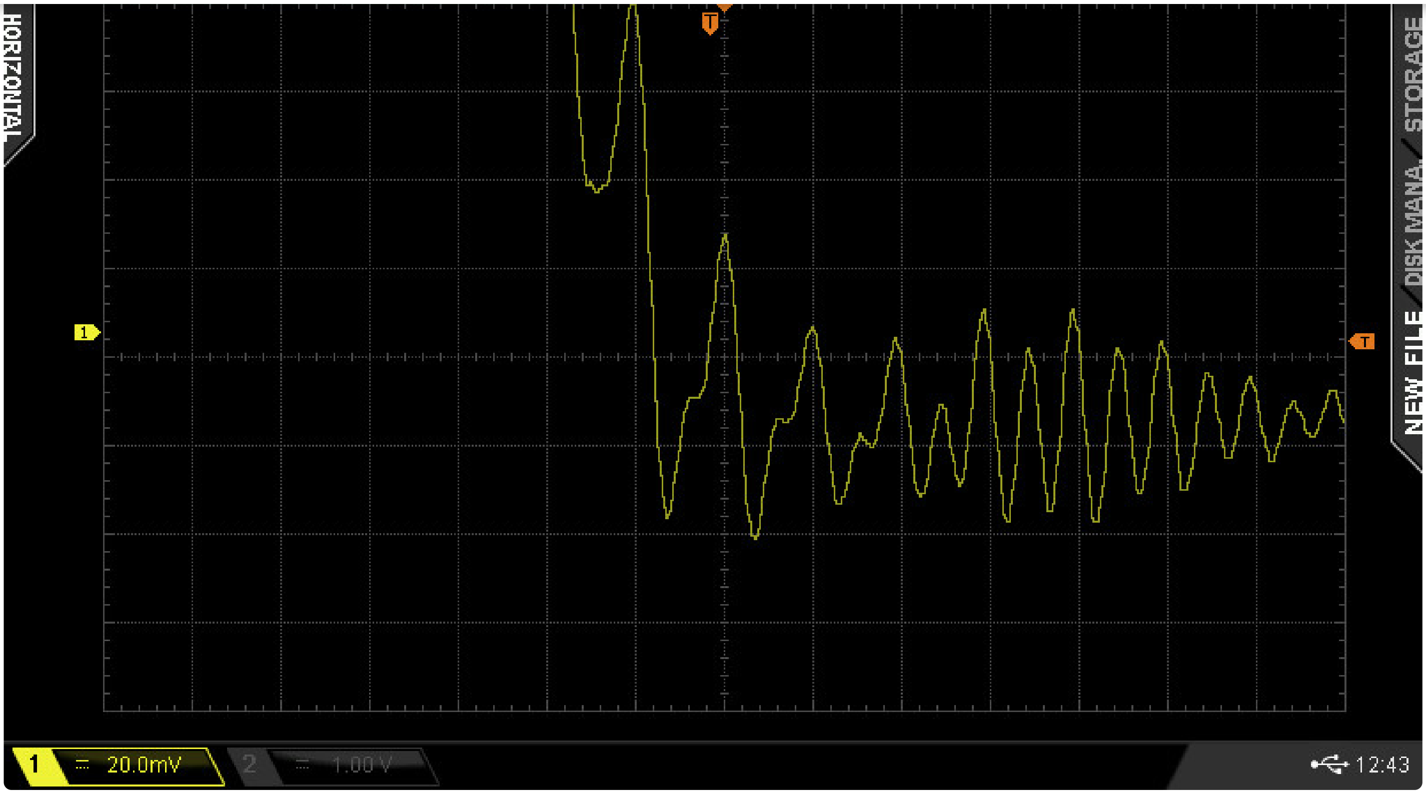
The following image is a picture of the original circuit that we used for modeling:



We captured some frames of the neuronal model output and obtained the following graphs. We were able to draw some similarities between the sample action potential and the circuit results. The stimulus (produced by Na+ ions in a nueron) and the resting state (produced by K+ ions in a neuron) are clearly visible.

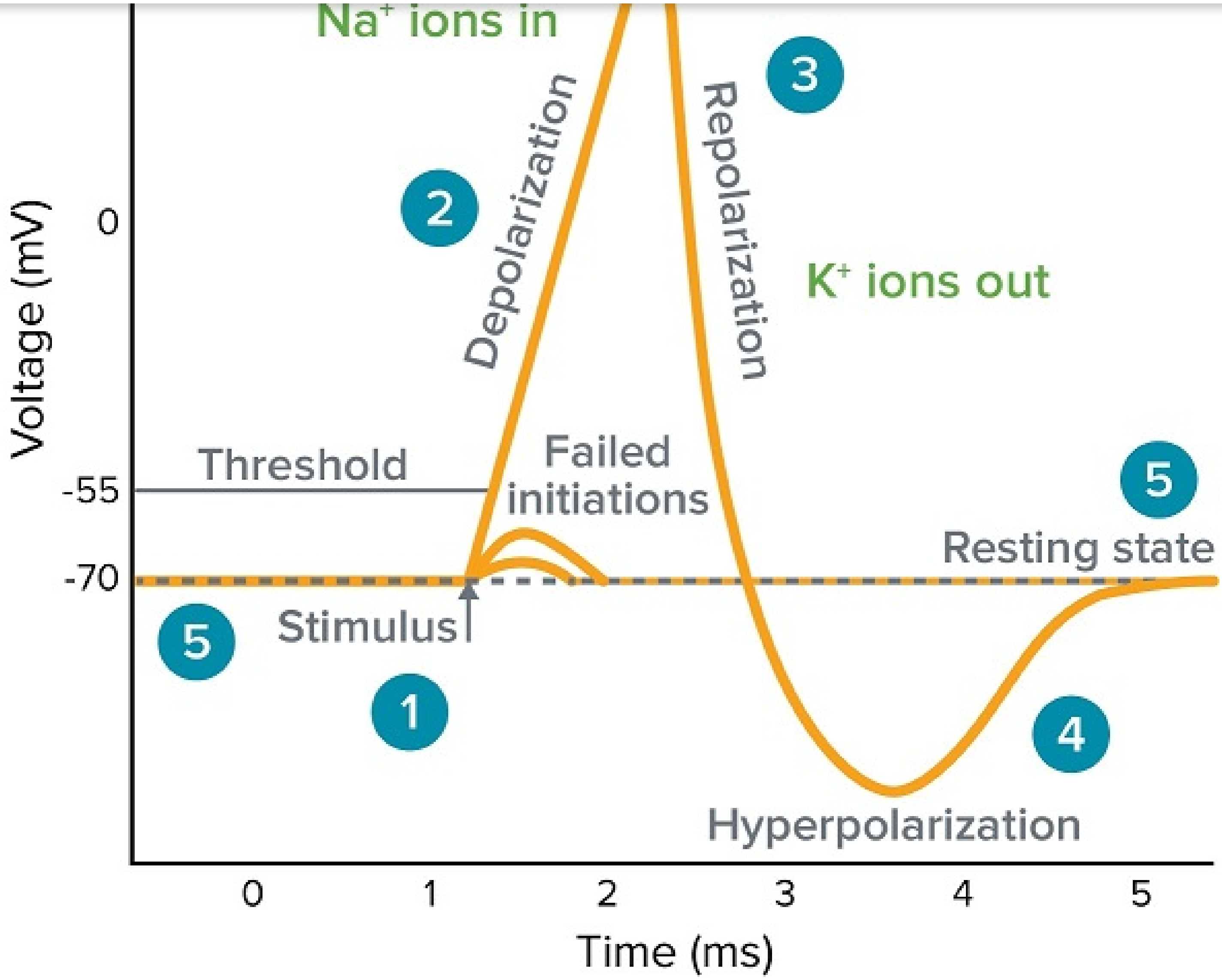
To test the circuit, we used a Tektronix AFG 3022B Dual Channel Function Generator to generate an input square wave of a 4Hz frequency and 5V of amplitude. We also used a Rigol DS2202 Digital Oscilloscope to measure the output of the wave passes. The screenshots that we present below are from the Oscilloscope.





Quoting the paper by Caltech, "As an aside, studies have shown that the dendrites of many types of neurons contain active ion channels (i.e. channels that change their conductance when a signal depolarizes or hyperpolarizes them). As a result, real dendrites can exhibit more complex behavior.. Still, the RC circuit model is often a reasonable approximation." Hence, we understand that this RC circuit is a big approximation of the neuronal behaviour.

Example Action Potential Plot

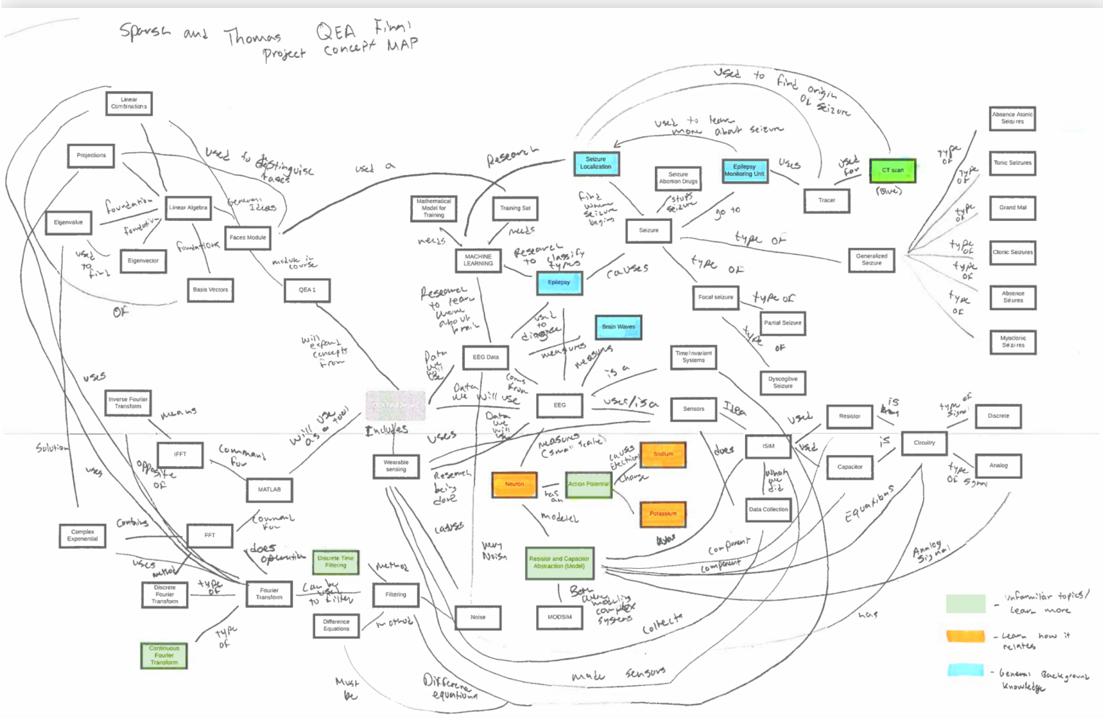


The source of this image can be found at the following link: Image Source.

Scientific Background Influence on Algorithm

Based on what we learned during our research, we found that seizures will cause an increased frequency and magnitude of signal in the EEG output. Due to this, we considered three different methods of classifying data as a seizure or not. The first was an algorithm based on principle component analysis. The second utilized a simple

Concept Map



Page 1 / 1

Olin College of Engineering, Needham MA
Email: sbansal, tjagielski (@olin.edu)

ANALYSIS

WE WERE ABLE TO CORRECTLY IDENTIFY IF A SEIZURE WAS PRESENT IN AN EEG SIGNAL 96.8% OF THE TIME!

Please note, all the code can be found in its respective subsection.

For our analysis, we considered three different methods of classification and compare their accuracies. The first algorithm we considered utilized principle component analyis and projecting new data into a basis set defined by training EEG data. Using this algorithm, we had a 88.4% accuracy rate. The second algorithm we considered was simply setting a threshold (defined by the training set) for the average value present in the Fourier transform of the data. This works because the magnitude of voltage increases with a seizure - compared to normal brainwaves. Using this algorithm we were accurate 96.8% of the time. The final algorithm we considered only looked at specific frequencies and smoothed the eigenvectors. Using this algorithm we were accurate 80% of the time with four or more points in the moving average filter.

The dataset contains the EEG signals for five different people. The data we used during this analysis can be found here.

DATASET

The following datasets contain data collected by medical professionals in hospitals using an EEG. The dataset contains 100 text files for each of the five persons, which each consists of 4096 samples collected at a sampling rate of 173.61 Hz. Each of the datasets only represents data from one individual. The data has been divided and selected for the most clean data from the entire dataset by doctors.

- Set Z contains data from a healthy person with their eyes open
- Set O contains data from a healthy person with their eyes closed
- Set N contains data from someone with epilepsy but does not contain any seizures
- Set F contains data from someone with epilepsy but does not contain any seizures
- Set S only contains seizure activity

  * NOTE: The only difference in sets N and F were that the data comes from different people and different regions of the brain.

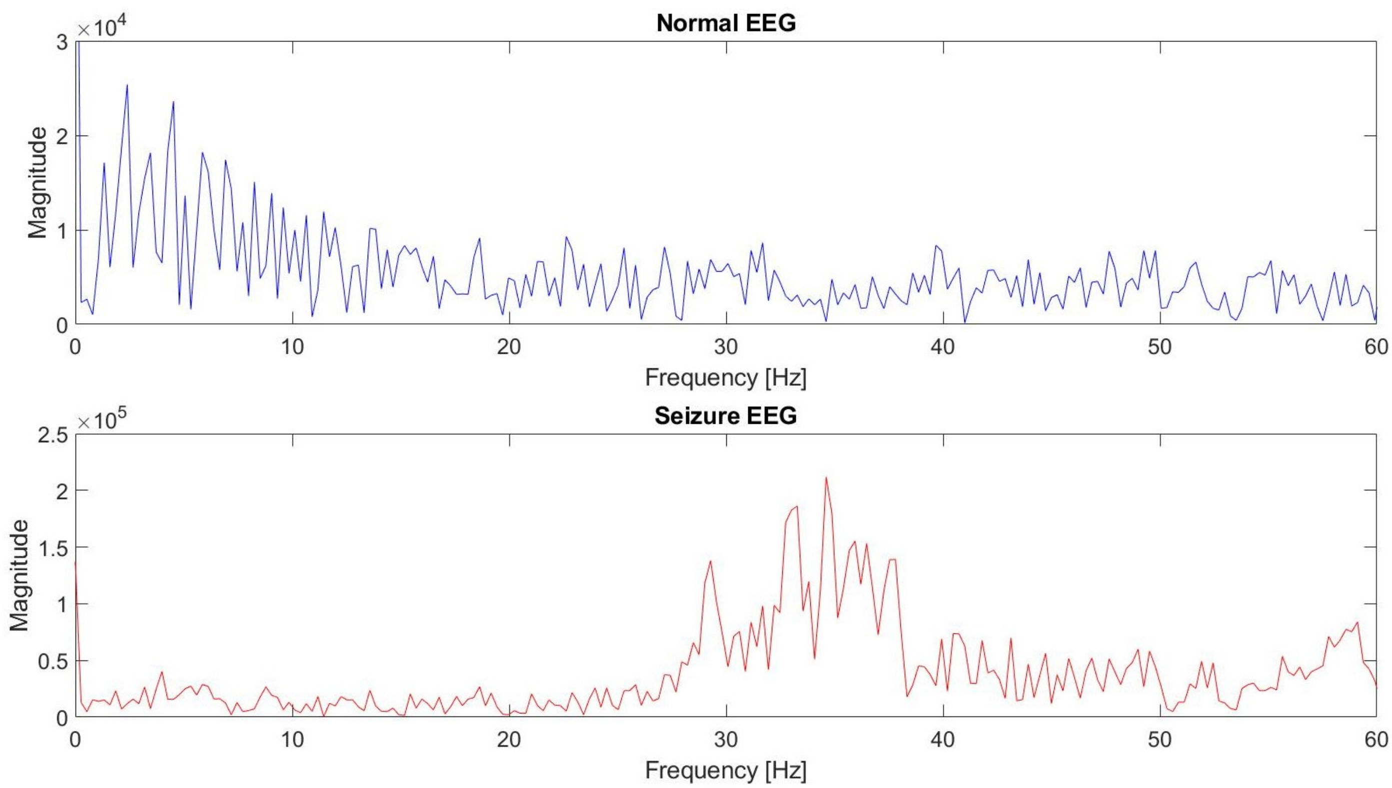
PREPARING DATA

The first step of our analysis is to put the data into column vectors in MATLAB. The code we used for this is found at the following links:

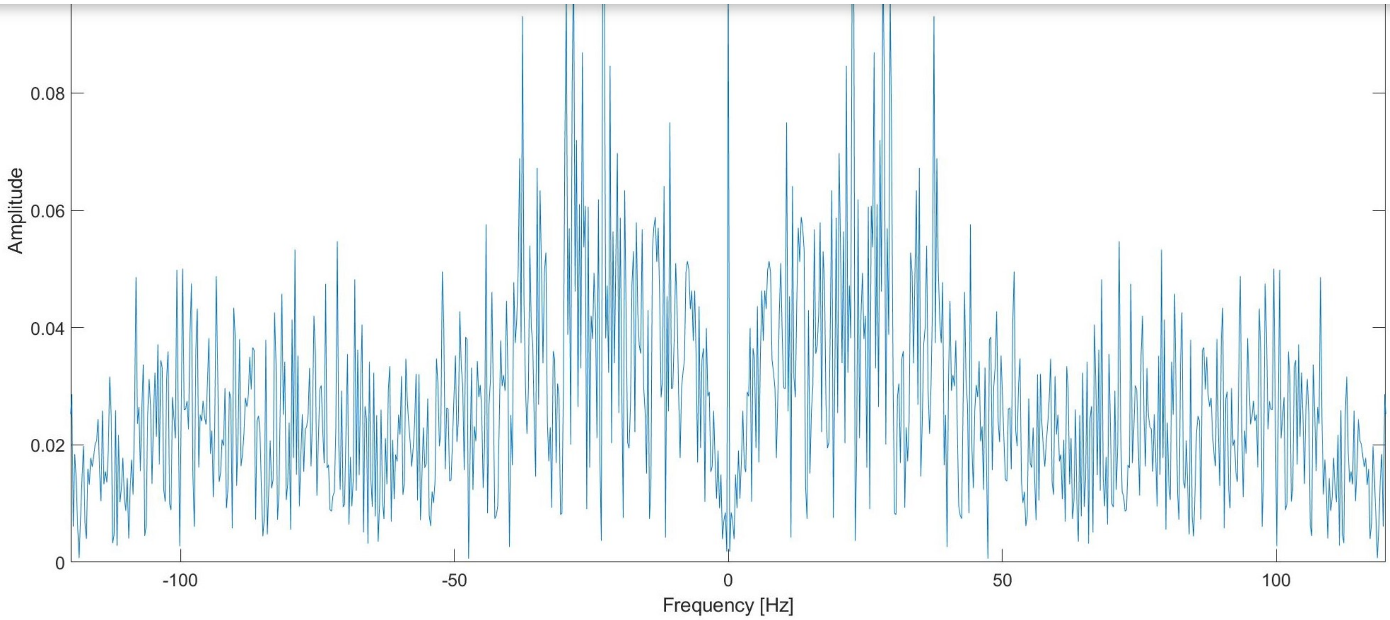
text file and each row is the time series data contained within an individual text file.

CLASSIFYING DATA: PRINCIPLE COMPONENT ANALYSIS

We separate the data into a test and training set. As the data has already been randomized by the collector of the data, we select the training set as the first 50 columns and the test set for the last 50. Once we have done this, we perform a discrete Fourier transform on the data to find out how much of each frequency is present. As we explained on the scientific background tab, an increased frequency of the signal and increased magnitude are two common characteristics of seizure EEG readings. This is depicted in the Fourier analysis of the following two signals:



We use an algorithm similar to the faces algorithm from QEA 1 Module 2 on the Fourier transformations of each of the training sets. We perform principal component analysis (PCA) on the training set to define a new set of basis vectors for the data. Each basis or Eigenvector represents a frequency that is the most variant across the dataset. A visualization of an eigenfrequency can be found below:



The spikes correspond with the frequencies that have the most variation across the training set. For example, we can see the highest spike occurring at 22 Hz. This means that there is lots of variation (because it is one of the first eigenfrequencies from PCA) at this frequency in the training set.

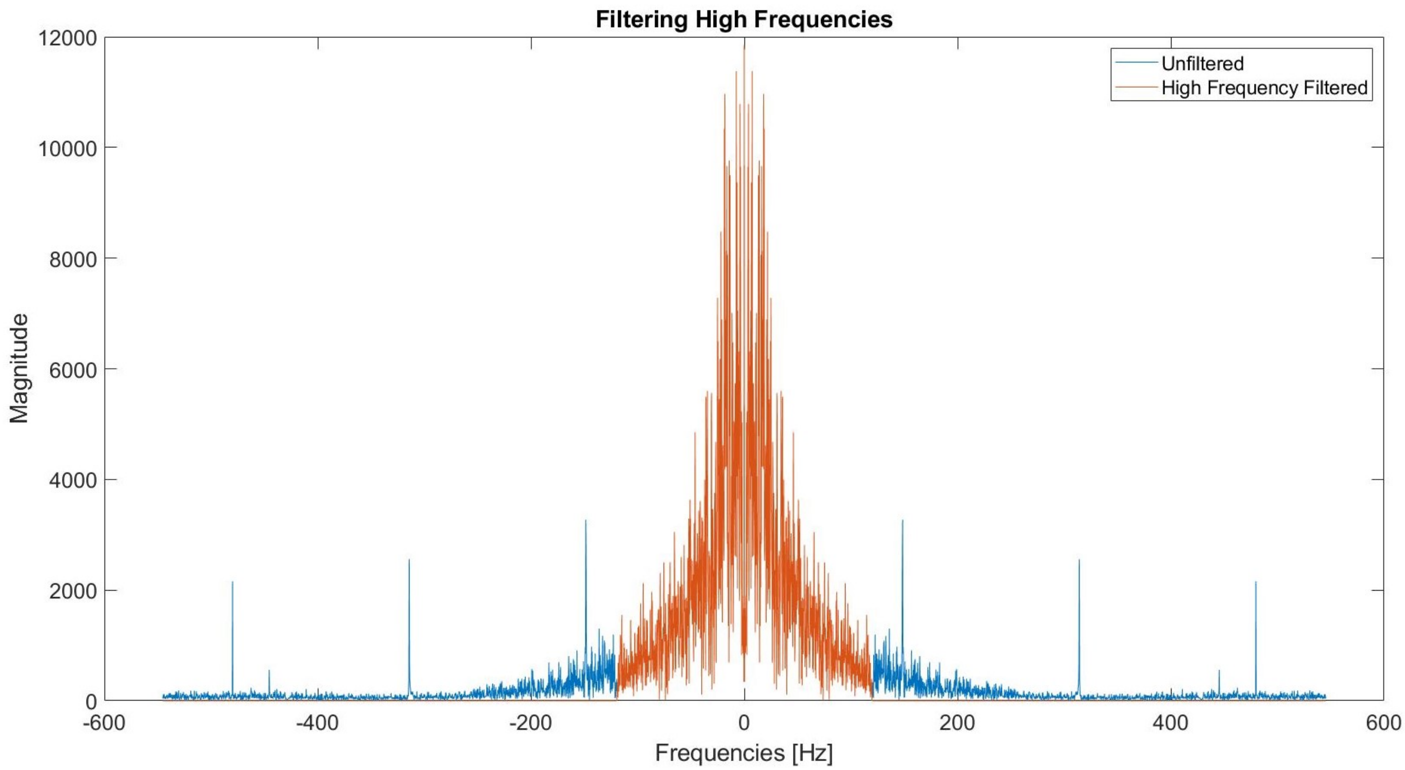
Using the test set, we project each of the Fouier transformations onto the basis vectors defined by the PCA of the training set. We then find the Euclidean Distance between the training and test sets in order to classify the test signal as either a seizure or not. The code used in this analysis can be found: **here.** Using this algorithm, we classified data correctly 88.4% of the time

CLASSIFYING DATA: FFT MAGNITUDE

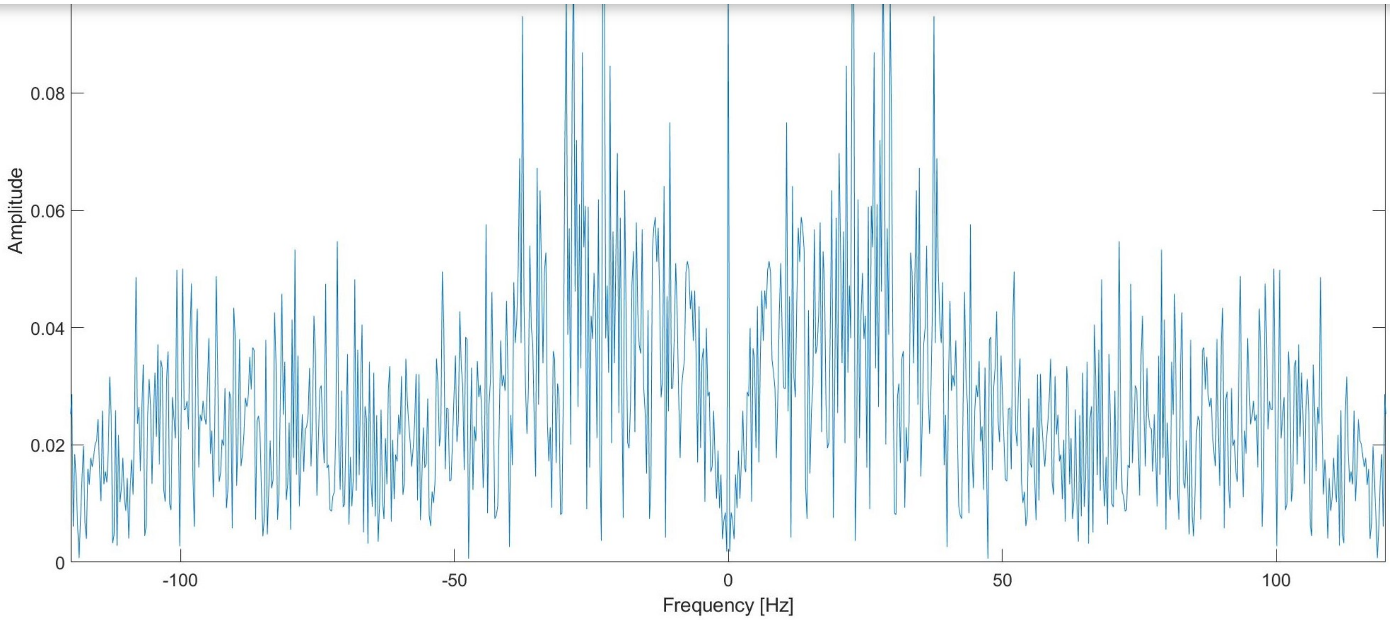
For this algorithm, we considered classifying the data based the average magnitude in the Fourier transfrom. We set the threshold by finding the average value of the Fourier transform. Once we have found the average of all data, we subtract a percentage of the average of only the seizure data. For the test data, we find the fourier transform, find the average magnitude and compare it to the threshold. If the average value is greater than the threshold, we classify the data as a seizure. If not, we classify the signal as normal brainwaves. The code we used in this algorithm can be found: **here.** This algorithm classified the data correctly 96.8% of the time.

CLASSIFYING DATA: FILTERING AND SMOOTHING DATA

To increase the ease of analysing the data, we use a moving average filter in order to smooth out some features of the data that we don't necessarily need to a full precision. We used difference equations on the Fourier transformed data to enable ourselves to focus on more useful aspects of the data. A difference demonstrates the filtering that we did:



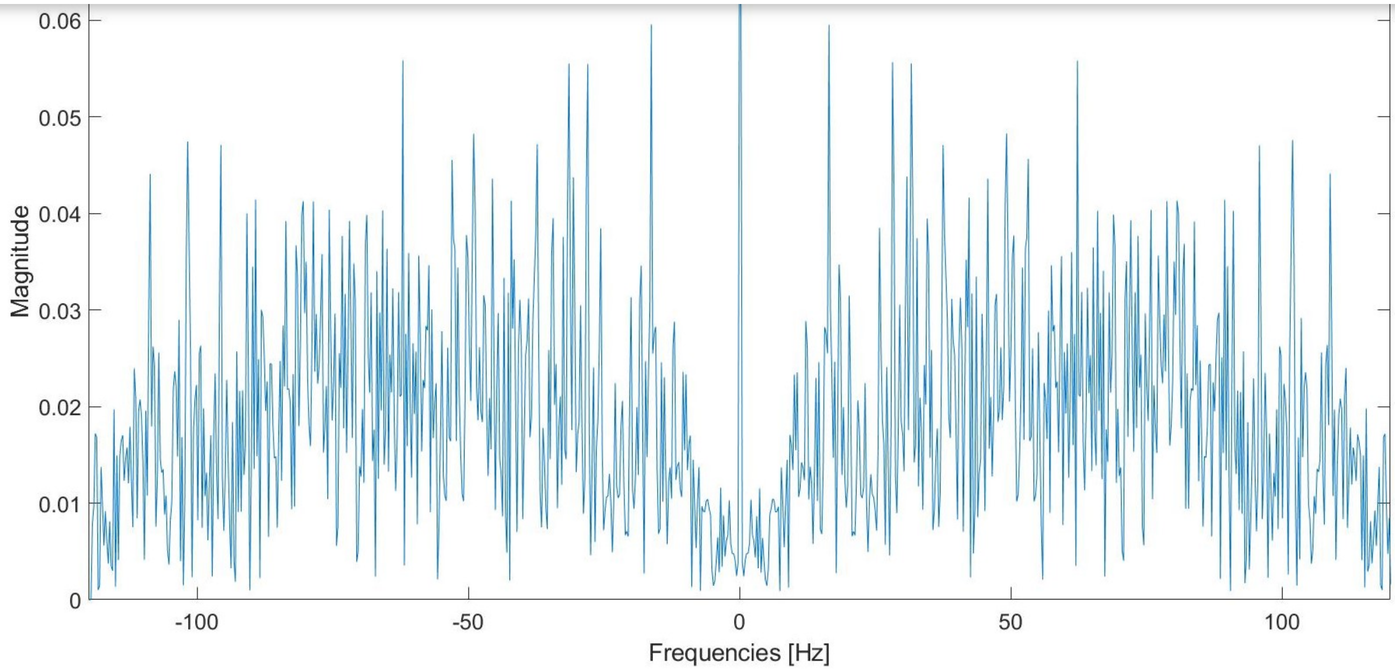
An unfiltered eigenfrequency looks like this:



We tested a two point moving average filter.

$$\text{Filtered}(x) = {}^{(x_n + x_{n+1})}/_{2}$$

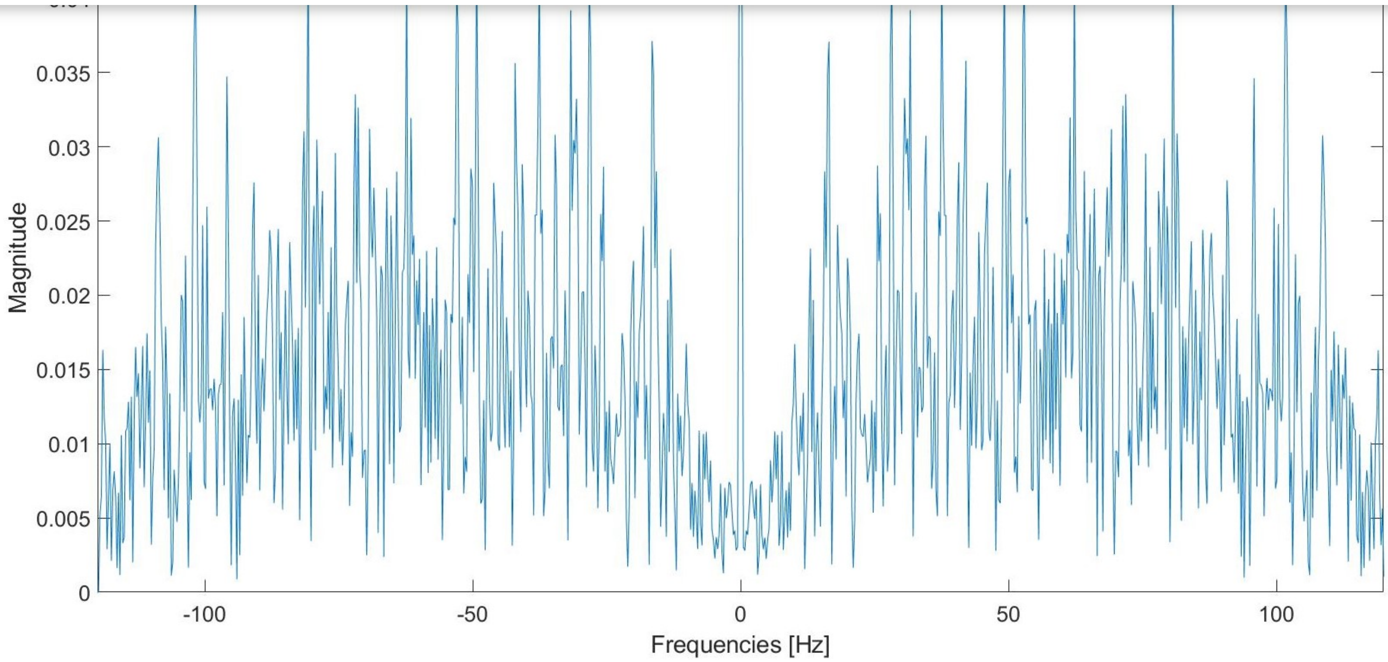
Post filtering, this graph appears as follows:



We tested a three point moving average filter.

$$\text{Filtered}(x) = {}^{(x_{n-1} + x_n + x_{n+1})}/_{3}$$

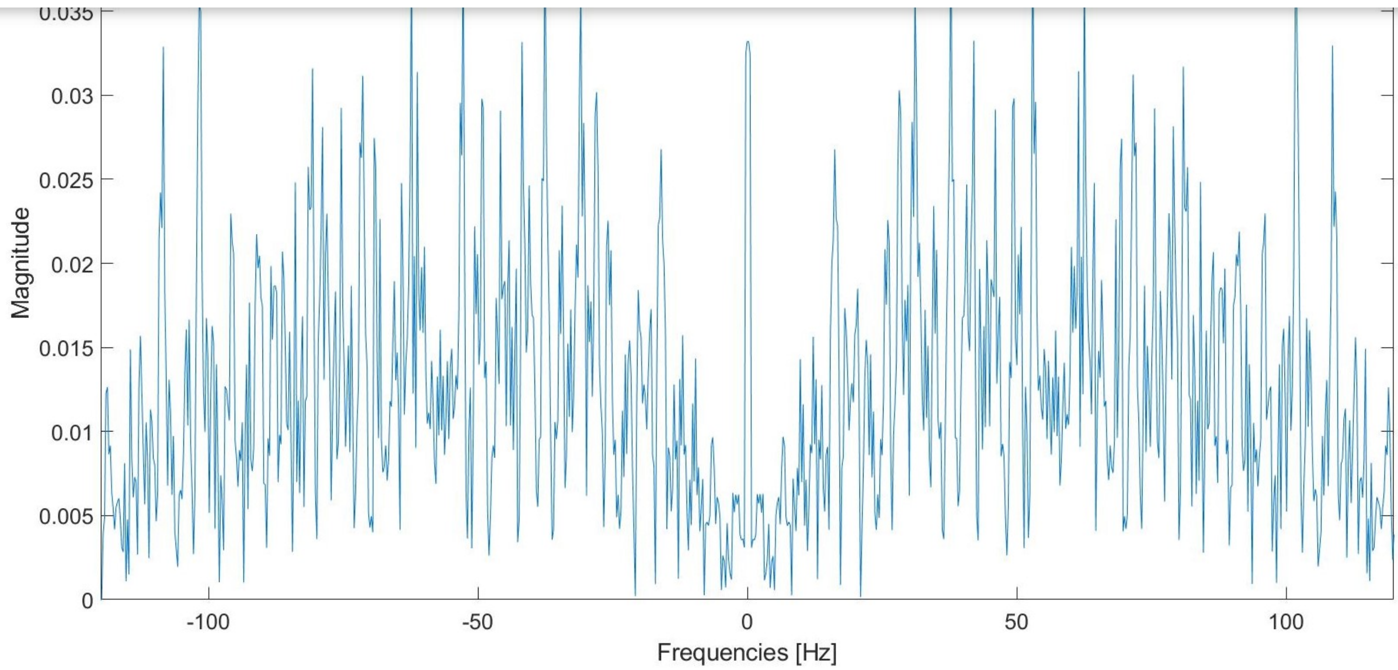
Post filtering, this graph appears as follows:



We tested a four point moving average filter.

$$\text{Filtered}(x) = {}^{(x_{n-2} + x_{n-1} + x_n + x_{n+1})}/_{4}$$

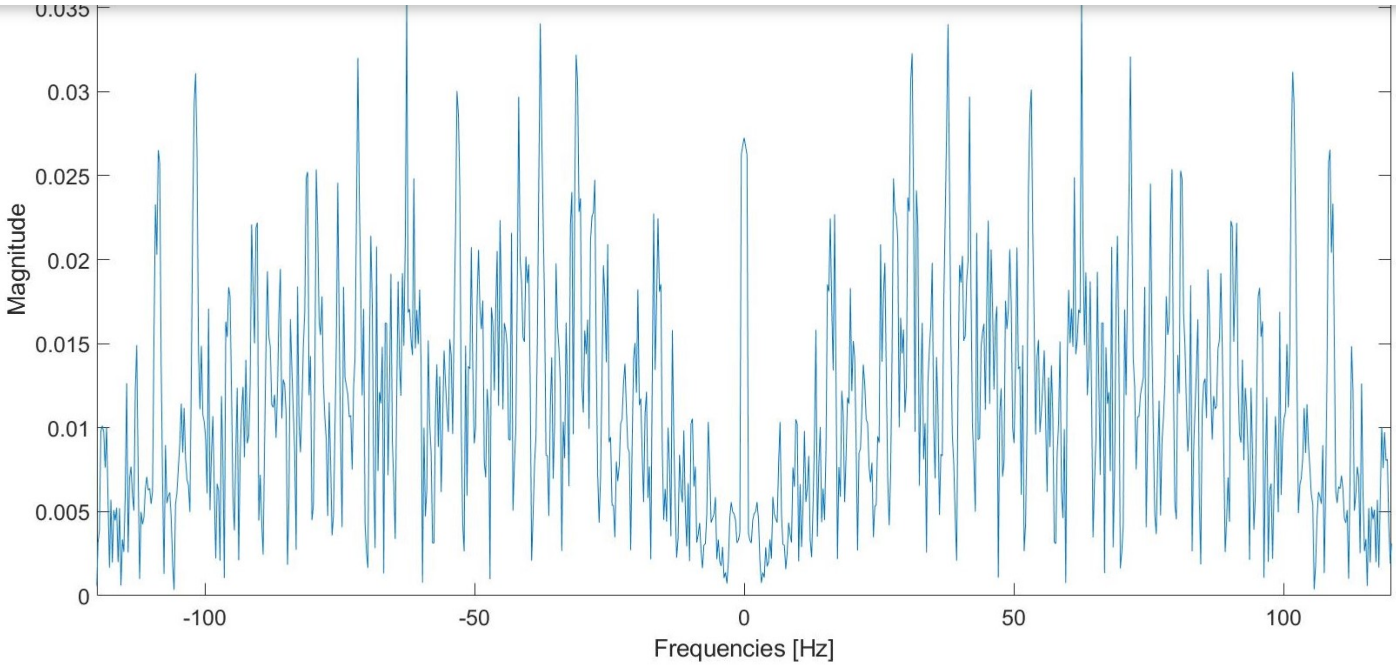
Post filtering, this graph appears as follows:



We tested a five point moving average filter.

$$\text{Filtered}(x) = {}^{(x_{n-2} + x_{n-1} + x_n + x_{n+1} + x_{n+2})}/_{5}$$

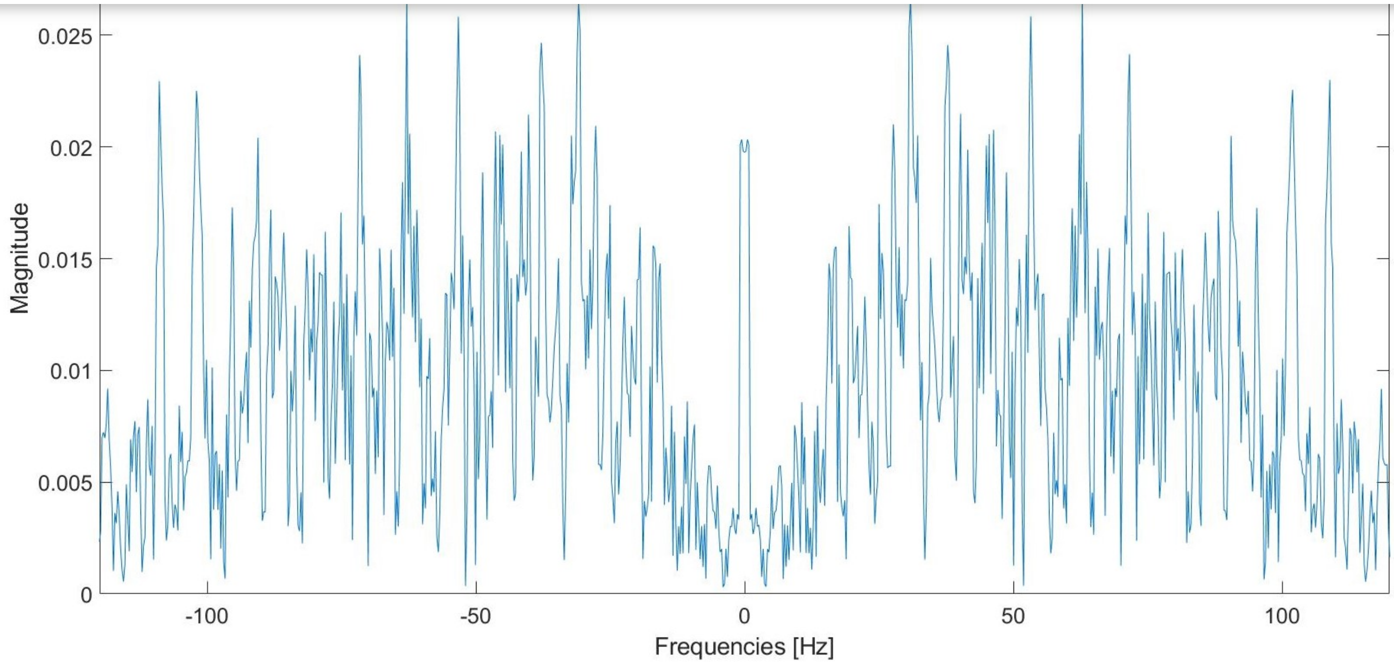
Post filtering, this graph appears as follows:



We used a seven point moving average filter.

$$\text{Filtered}(x) = {}^{(x_{n-3} + x_{n-2} + x_{n-1} + x_n + x_{n+1} + x_{n+2} + x_{n+3})}/_{7}$$

Post filtering, this graph appears as follows:



The function used to zero high frequencies can be found **here.** The algorithm code used in this analysis can be found: **here.** Using this algorithm, we acheived the following accuracy for various number of points included in the moving average filter.

Number of Points in the Moving Average Filter	Accuracy
1	88.4%
2	82%
3	80.4%
4	80%
5	80%
7	80%

CONTACT

Suggestions? Potential Applications? Drop a note!

FUTURE DEVELOPMENTS

Currently, the Fourier analysis we are performing projects the signal on many cosine waves at different frequencies to find the frequencies present in the signal. However, many of the common EEG waves that appear are not cosine waves.

To increase the accuracy and confidence in our algorithm, an interesting future development would be to perform Fourier analysis with common seizure EEG waves as opposed to cosine waves. In doing this, we would be able to sweep across the waveforms that you would expect to see at various frequencies in order to have a more accurate projection.

As an example, this could be done with a wavelet transform. Some resources to information on wavelet transforms can be found below.

Wavelet Transform

Other Transformations

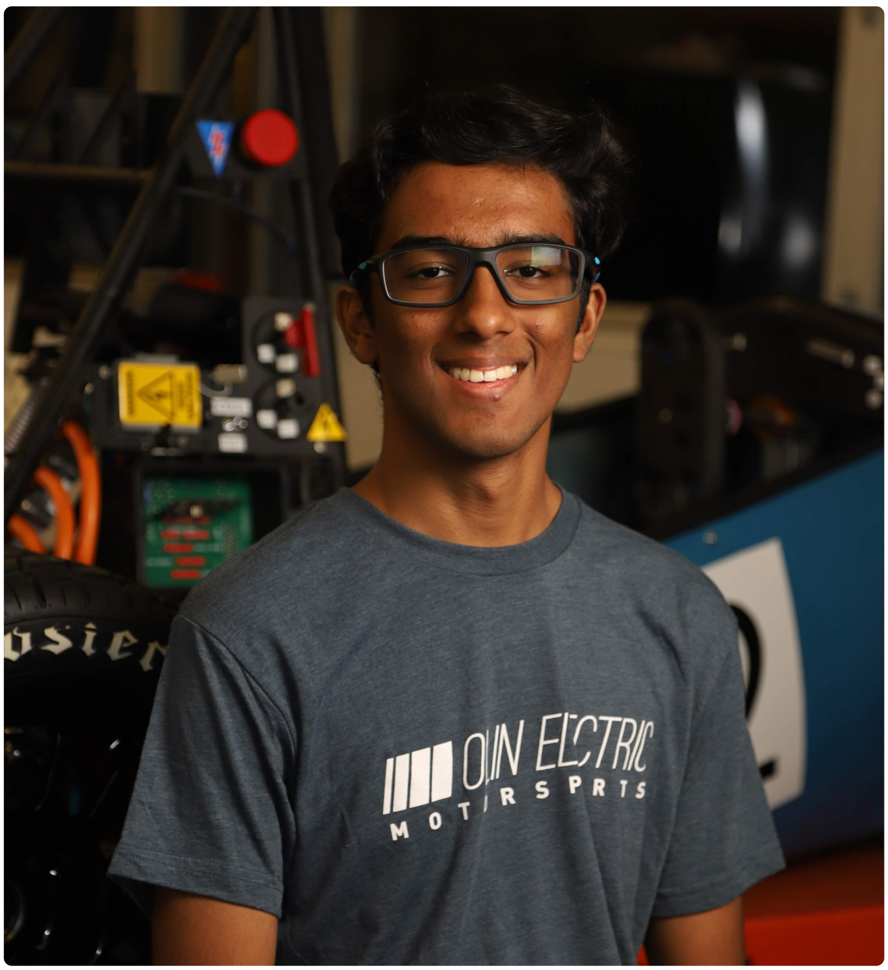
CONTACT

Suggestions? Potential Applications? Drop a note!

Olin College of Engineering, Needham MA
Email: sbansal@olin.edu

AUTHORS

Sparsh Bansal



Thomas Jagielski

SPARSH BANSAL

Electrical and Computer Engineer 22'

CONTACT

Suggestions? Potential Applications? Drop a note!

Olin College of Engineering, Needham MA

Phone: +781-708-1497

Email: sbansal, tjagielski (@olin.edu)

Name

Email

Message

SEND

Powered by w3.css

ACKNOWLEDGMENTS

We would like to use this space to thank everyone at or outisde of Olin College who guided us through this project for their resourcefullness. A special thanks to Jack Jagielski for their support on the seizure mechanisms and methods from the Mayo Clinic, and to Dr. Sam Michalka for her support on EEG sensing, instrumentation, and experimentation from the Human Augmentation Lab at Olin College. Some other references are listed below under topics as headers:

A. Action potential

- "2-Minute Neuroscience: Action Potential," *YouTube*. [Online]. Available: https://www.youtube.com/watch?v=W2hHt_PXe5o. [Accessed: 17-Nov-2019].

- "Neuron action potential description," *Khan Academy*. [Online]. Available: https://www.khanacademy.org/science/health-and-medicine/nervous-system-and-sensory-infor/neuron-membrane-potentials-topic/v/neuron-action-potential-description. [Accessed: 17-Nov-2019]

B. Resistor and Capacitor Abstraction (Neuron Model)

- K. M. Dabrowski, D. J. Castaño, and J. L. Tartar, "Basic neuron model electrical equivalent circuit: an undergraduate laboratory exercise,*" Journal of undergraduate neuroscience education : JUNE : a publication of FUN, Faculty for Undergraduate Neuroscience*, 15-Oct-2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3852871/. [Accessed: 17-Nov-2019].

- "Applied Neuroscience,*" SHP Applied Neuroscience*. [Online]. Available: http://www.columbia.edu/cu/appliedneuroshp/. [Accessed: 17-Nov-2019].

- Wagenaar, Daniel A., and Kevin Briggman. *"Electronic Circuits for Electrophysiologist."* http://www.its.caltech.edu/, California Institute of Technology, http://www.its.caltech.edu/~daw/assets/circuits.pdf.

C. Discrete Time Filtering

- *Discrete-Time Signals & Digital Filtering*. [Online]. Available: https://www.music.mcgill.ca/~gary/618/week1/signals.html. [Accessed: 17-Nov-2019].

- "Digital filter," *Wikipedia*, 16-Nov-2019. [Online]. Available: https://en.wikipedia.org/wiki/Digital_filter. [Accessed: 17-Nov-2019].

- Oppenheim and A. V., "Lecture 8: Continuous-Time Fourier Transform," MIT OpenCourseWare, Massachusetts Institute of Technology. [Online]. Available: https://ocw.mit.edu/resources/res-6-007-signals-and-systems-spring-2011/video-lectures/lecture-8-continuous-time-fourier-transform/. [Accessed: 17-Nov-2019].

- "Lecture 8, Continuous-Time Fourier Transform | MIT RES.6.007 Signals and Systems, Spring 2011," *YouTube*. [Online]. Available: https://www.youtube.com/watch?v=3UkGd3LK2NY. [Accessed: 17-Nov-2019].

E. Neuron

- "Neuron Action Potentials: The Creation of a Brain Signal," *Khan Academy*. [Online]. Available: https://www.khanacademy.org/test-prep/mcat/organ-systems/neuron-membrane-potentials/a/neuron-action-potentials-the-creation-of-a-brain-signal. [Accessed: 17-Nov-2019].

- "The Synapse," *Khan Academy*. [Online]. Available: https://www.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/the-synapse. [Accessed: 17-Nov-2019].

F. Sodium

- "Neuron Action Potentials: The Creation of a Brain Signal," *Khan Academy*. [Online]. Available: https://www.khanacademy.org/test-prep/mcat/organ-systems/neuron-membrane-potentials/a/neuron-action-potentials-the-creation-of-a-brain-signal. [Accessed: 17-Nov-2019].

- "The Synapse," *Khan Academy*. [Online]. Available: https://www.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/the-synapse. [Accessed: 17-Nov-2019].

G. Potassium

- "Neuron Action Potentials: The Creation of a Brain Signal," *Khan Academy*. [Online]. Available: https://www.khanacademy.org/test-prep/mcat/organ-systems/neuron-membrane-potentials/a/neuron-action-potentials-the-creation-of-a-brain-signal. [Accessed: 17-Nov-2019].

- "The Synapse," *Khan Academy*. [Online]. Available: https://www.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/the-synapse. [Accessed: 17-Nov-2019].

- I. Chang-Hwan, *Computational EEG Analysis: Methods and Applications*. S.l.: SPRINGER, 2019.
- W. J. Freeman and R. Q. Quiroga, *Imaging Brain Function With EEG Advanced Temporal and Spatial Analysis of Electroencephalographic Signals*. New York, NY: Springer New York, 2013.

I. Seizure Localization

- K. Tufenkjian and H. O. Lüders, "Seizure Semiology: Its Value and Limitations in Localizing the Epileptogenic Zone," *Journal of Clinical Neurology*, vol. 8, no. 4, p. 243, 2012.
- A. Arain, [Online]. Available: https://www.mc.vanderbilt.edu/documents/neurology/files/Focal%20epilepsies%20Arain.pdf.
- M. M. Jan and J. P. Girvin, "Seizure Semiology: Value in Identifying Seizure Origin," *Canadian Journal of Neurological Sciences / Journal Canadien des Sciences Neurologiques*, vol. 35, no. 1, pp. 22–30, 2008.

J. Epilepsy Monitoring Unit

- "Epilepsy Monitoring Units," *American Epilepsy Society*, 21-Jan-2014. [Online]. Available: https://www.aesnet.org/clinical_resources/epilepsy_monitoring_units.

K. Eigenvector Classification Algorithm

- Matthew Turk and Alex Pentland, "Eigenfaces for Recognition", The MIT Media Laboratory

L. Brainwaves

- "EEG Introduction," *The McGill Physiology Virtual Lab.*. [Online]. Available: https://www.medicine.mcgill.ca/physio/vlab/biomed_signals/eeg_n.htm.

CONTACT

Email: sbansal@olin.edu

Contents


```
function VarName1 = importfile(filename, dataLines)
```

```
%IMPORTFILE Import data from a text file
%  VARNAME1 = IMPORTFILE(FILENAME) reads data from text file FILENAME
%  for the default selection.  Returns the data as column vectors.
%
%  VARNAME1 = IMPORTFILE(FILE, DATALINES) reads data for the specified
%  row interval(s) of text file FILENAME. Specify DATALINES as a
%  positive scalar integer or a N-by-2 array of positive scalar integers
%  for dis-contiguous row intervals.
%
%  Example:
%  VarName1 = importfile("C:\Users\tjagielski\Documents\Projects\School\Sophomore - Semester 1\QEA 2\Module 2\EEG Data\F.txt", [1, Inf]);
%
%  See also READTABLE.
%
% Auto-generated by MATLAB on 28-Nov-2019 17:38:14
```

Input handling

```
% If dataLines is not specified, define defaults
if nargin < 2
    dataLines = [1, Inf];
end
```

Setup the Import Options

```
opts = delimitedTextImportOptions("NumVariables", 1);

% Specify range and delimiter
opts.DataLines = dataLines;
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = "VarName1";
opts.VariableTypes = "double";
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Import the data
tbl = readtable(filename, opts);
```

Convert to output type

```
VarName1 = tbl.VarName1;
```

```
end
```

Published with MATLAB® R2019a

```
close all
clear all
clc

% Define variable as the dataset letter being used
variable = "F";

% Set the number of files as NN
NN = 100;
for i = 1:NN
    number = string(i); % Convert the index number to a string
    if strlength(number) == 1 % Compare the length of the index
        number = "00" + number; % If the index is length one, add two zeros in front
                                % This is how the dataset is defined
    end
    if strlength(number) == 2 % Compare the length of the index
        number = "0" + number; % If the index is length two, add one zero in front
                               % This is how the dataset is defined
    end
    name = variable + number + ".txt"; % Convert the filename to a string
    % Set the filepath
    vectorname = "C:\Users\tjagielski\Documents\Projects\School\Sophomore - Semester 1\QEA 2\Module 2\EEG Data\" + variable + "\" + name;
    vector = importfile(vectorname,[1, Inf]); % Import the current text file defined by the loop iteration
    output(:,i) = vector; % Add the vector to an output matrix of the entire dataset
end
```

Published with MATLAB® R2019a

```
% 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

```
% 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

% Training sets defined by the first 50 columns
F = F(:,1:50);
N = N(:,1:50);
O = O(:,1:50);
S = S(:,1:50);
Z = Z(:,1:50);

Fs = 173.61; % Define the sampling frequency
N_shift = length(F); % Set the length of the shifted dataset
% Define the frequency_shifted vector
frequencies_shifted = (linspace(-pi*Fs, Fs*(pi - (2*pi)/N_shift), N_shift) + (Fs*pi)/(N_shift)*mod(N_shift, 2))';

% Find the fftshift for dataset F
ffft = fft(F);
ffft = fftshift(ffft);

% Find the fftshift for dataset N
nfft = fft(N);
nfft = fftshift(nfft);

% Find the fftshift for dataset O
offt = fft(O);
offt = fftshift(offt);

% Find the fftshift for dataset S
sfft = fft(S);
sfft = fftshift(sfft);

% Find the fftshift for dataset Z
zfft = fft(Z);
zfft = fftshift(zfft);

% Find the average of all the datasets
f = mean(abs(ffft));
n = mean(abs(nfft));
o = mean(abs(offt));
s = mean(mean(abs(sfft)));
z = mean(abs(zfft));

% Define vector that contains the non-seizure average EEG readings
nons = mean([f,n,o,z]);
% Define the threshold as the mean of seizure and non-seizure averages
% minus an offset to offset the threshold
threshold = mean([nons,s])-(0.2*s);
```

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

% Test sets defined by the last 50 columns
F2 = F(:,51:end);
N2 = N(:,51:end);
O2 = O(:,51:end);
S2 = S(:,51:end);
Z2 = Z(:,51:end);

% Find the fftshifted for the test sets
ffft = fft(F2);
ffft = fftshift(ffft);

nfft = fft(N2);
nfft = fftshift(nfft);

offt = fft(O2);
offt = fftshift(offt);

sfft = fft(S2);
sfft = fftshift(sfft);

zfft = fft(Z2);
zfft = fftshift(zfft);

% Find the averages of the fft magnitude
f = mean(abs(ffft));
n = mean(abs(nfft));
o = mean(abs(offt));
s = mean(abs(sfft));
z = mean(abs(zfft));

% Define a new matrix that contains all the averages
mat = [s,f,n,o,z];
% Initialize the counter
counter = 0;
for l=1:length(mat)
    if (mat(l) >= threshold) && (l <= 50) % If the test is classified as a seizure reading
        counter = counter + 1; % Add one to the counter
    elseif (51 <= l) && (mat(l) <= threshold) % If the test is classified as a non-seizure reading
        counter = counter + 1; % Add one to the counter
    end
end
% Find the accuracy
accuracy = counter / length(mat) * 100
```

```
accuracy =

   96.8000
```

Published with MATLAB® R2019a

```
function rangedfft = Rangefinder(transform, low, high)
% Function used to zero high frequency signals
% INPUTS: transform - the FFT vector
%         low - lower index threshold
%         high - higher index threshold
% OUTPUT: rangedfft - FFT vector with zeroed values lower than "low" and
%                     higher than "high"
for l=1:max(size(transform))
    if l <= low % Compare index to the lower value
        transform(l) = 0; % If the index is below the "low" index, zero it
    end
    if l >= high % Compare index to the higher value
        transform(l) = 0; % If the index is larger the "high" index, zero it
    end
end
rangedfft = transform; % Set the output equal to the filtered FFT
end
```

Published with MATLAB® R2019a

```
% 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

% Training sets defined by the first 50 columns
F = F(:,1:50);
N = N(:,1:50);
O = O(:,1:50);
S = S(:,1:50);
Z = Z(:,1:50);

Fs = 173.61; % Define the sampling frequency
N_shift = length(F); % Set the length of the shifted dataset
% Define the frequency_shifted vector
frequencies_shifted = (linspace(-pi*Fs, Fs*(pi - (2*pi)/N_shift), N_shift) + (Fs*pi)/(N_shift)*mod(N_shift, 2))';

% Define the lower and higher indicies to filter
% These indicies will filter out signals lager than 120 Hz
lower_filter = 1600;
higher_filter = 2499;

% Find the shifted Fourier Transform
ffft = fft(F);
ffft = fftshift(ffft);
% Filter out frequencies higher than 120 Hz
ffft_filtered = zeros(size(ffft));
for k=1:size(ffft,2)
    ffft_filtered(:,k) = Rangefinder(ffft(:,k),lower_filter,higher_filter);
end
ffft = ffft_filtered;

% Find the shifted Fourier Transform
nfft = fft(N);
nfft = fftshift(nfft);
% Filter out frequencies higher than 120 Hz
nfft_filtered = zeros(size(nfft));
for k=1:size(nfft,2)
    nfft_filtered(:,k) = Rangefinder(nfft(:,k),lower_filter,higher_filter);
end
nfft = nfft_filtered;

% Find the shifted Fourier Transform
offt = fft(O);
offt = fftshift(offt);
% Filter out frequencies higher than 120 Hz
offt_filtered = zeros(size(offt));
for k=1:size(offt,2)
    offt_filtered(:,k) = Rangefinder(offt(:,k),lower_filter,higher_filter);
end
offt = offt_filtered;

% Find the shifted Fourier Transform
sfft = fft(S);
sfft = fftshift(sfft);
% Filter out frequencies higher than 120 Hz
```

```
sfft_filtered = zeros(size(sfft));
for k=1:size(sfft,2)
    sfft_filtered(:,k) = Rangefinder(sfft(:,k),lower_filter,higher_filter);
end
sfft = sfft_filtered;

% Find the shifted Fourier Transform
zfft = fft(Z);
zfft = fftshift(zfft);
% Filter out frequencies higher than 120 Hz
zfft_filtered = zeros(size(zfft));
for k=1:size(zfft,2)
    zfft_filtered(:,k) = Rangefinder(zfft(:,k),lower_filter,higher_filter);
end
zfft = zfft_filtered;

% Make a matrix with all the data
data = [sfft,nfft,offt,ffft,zfft];
% Find the principle component analysis for the matrix defined by all the
% data
[U,SS,VV] = svd(data,'econ');
% Find the linear combination of training data into a new basis set defined
% by the eigenvectors of the data
train_weights = U' * data;

% Averaging filter applied to the eigenvectors
for l=1:size(U,2)
    U(:,l) = movmean(U(:,l),7);
end
```

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

% Test sets defined by the last 50 columns
F2 = F(:,51:end);
N2 = N(:,51:end);
O2 = O(:,51:end);
S2 = S(:,51:end);
Z2 = Z(:,51:end);

% Find the shifted Fourier Transform
ffft = fft(F2);
ffft = fftshift(ffft);
% Filter out frequencies higher than 120 Hz
ffft_filtered = zeros(size(ffft));
for k=1:size(ffft,2)
    ffft_filtered(:,k) = Rangefinder(ffft(:,k),lower_filter,higher_filter);
end
ffft = ffft_filtered;

% Find the shifted Fourier Transform
nfft = fft(N2);
nfft = fftshift(nfft);
% Filter out frequencies higher than 120 Hz
nfft_filtered = zeros(size(nfft));
for k=1:size(nfft,2)
    nfft_filtered(:,k) = Rangefinder(nfft(:,k),lower_filter,higher_filter);
```

```
end
nfft = nfft_filtered;

% Find the shifted Fourier Transform
offt = fft(O2);
offt = fftshift(offt);
% Filter out frequencies higher than 120 Hz
offt_filtered = zeros(size(offt));
for k=1:size(offt,2)
    offt_filtered(:,k) = Rangefinder(offt(:,k),lower_filter,higher_filter);
end
offt = offt_filtered;

% Find the shifted Fourier Transform
sfft = fft(S2);
sfft = fftshift(sfft);
% Filter out frequencies higher than 120 Hz
sfft_filtered = zeros(size(sfft));
for k=1:size(sfft,2)
    sfft_filtered(:,k) = Rangefinder(sfft(:,k),lower_filter,higher_filter);
end
sfft = sfft_filtered;

% Find the shifted Fourier Transform
zfft = fft(Z2);
zfft = fftshift(zfft);
% Filter out frequencies higher than 120 Hz
zfft_filtered = zeros(size(zfft));
for k=1:size(zfft,2)
    zfft_filtered(:,k) = Rangefinder(zfft(:,k),lower_filter,higher_filter);
end
zfft = zfft_filtered;

% Define matrix for the test data
test = [sfft,ffft,nfft,offt,zfft];
% Project the test vectors into the eigenspace
test_weights = U' * test;

counter = 0;
for l=1:length(test_weights(1,:))
    % Find Euclidean distance to find nearest image
    [dist,index] = min(vecnorm(test_weights(:,l) - train_weights));
    % Check if the test vector is correctly identified as a seizure
    if l <= 50 && (1 <= index) && (index <= 50)
        counter = counter + 1;
    % Check if the test vector is correctly identified as a seizure
    elseif (51 <= l) && (51 <= index)
        counter = counter + 1;
    end
end
% Compute the accuracy of the algorithm
accuracy = counter / length(test_weights(:,1)) * 100
```

```
accuracy =

    80
```

Published with MATLAB® R2019a