Department of Electrical Engineering and Computing Systems (DEECS)

Classifying Digits Presented to User via CNN Running on EEG Brain Activity

Anastasiya Chapko

April 28, 2020

Submitted in partial fulfillment of the degree of

Bachelor of Science in

Computer Engineering

Advisor: Dr. Rashmi Jha

Table of Contents

Table of Figures

Abstract

Hypothesis: Electroencephalography and Artificial Neural Networks can be combined to read in a user's EEG-based brain activity and then to correctly classify that activity.

Goal: This research project aims to combine EEG (electroencephalography) and ANNs (artificial neural networks) by reading in a user's EEG-based brain activity and using an ANN to correctly classify that activity. This specific application aims to classify EEG data of a user being presented with digits (0-9) and letters (A-J).

Process: The project goals are accomplished by building an EEG headset capable of collecting data, generating a labeled dataset (EEG activity is the data, character being presented is the label), and creating ANNs to analyze the labeled dataset.

Results: The EEG headset based on the UltraCortex III from OpenBCI was successfully built. A data collection protocol was created, programs were coded to facilitate this data collection, and the dataset was successfully generated (3160 samples total, 158 samples/character). Several ANNs were created, and these networks were capable of learning and of overfitting on the data, but the classification on test data did not reach accuracy levels beyond chance (more time needs to be devoted in trying different networks and manipulating the data).

Future work: Tasks which are recommended for continuing this project include adjusting network parameters of existing CNNs, trying a wider variety of neural network architectures, trying data mining techniques, extracting more features in different ways from the existing data, collecting more digit and letter EEG data, altering data collection process.

Introduction

Problem/Need (Hypothesis)

General: Electroencephalography and Artificial Neural Networks can be combined to read in a user's EEG-based brain activity and then to correctly classify that activity.

Final Application: Collecting data by reading in a user's EEG-based brain activity as they are presented with 10 different digits (0-9) and 10 different letters (A-J), building and training an artificial neural network on the collected data, and then classifying which character or what type of character the user was presented with based on a sample EEG recording alone.

Expected Results: I expect to be able to differentiate between letters and numbers, I expect there to be a correlation between A, B, C, D and 1, 2, 3, 4, respectively, and I expect that scoping to fewer digits or letters will yield better accuracy than including all the characters.

[*Previous (Discontinued) Applications*: Concentration Improvement, Direction Classification]

Solution

I propose a solution which is capable of presenting the user with a character (a digit 0-9 or a letter A-J), of collecting the user's EEG-based brain activity during that time, and of processing that information to 'guess' what character the user saw. This solution will be accomplished by creating an EEG headset capable of collecting EEG activity, generating a labeled dataset through a defined data collection process, and coding ANNs for classification of EEG activity.

Credibility

<u>Student (Anastasiya Chapko):</u>

This project requires backgrounds in neuroscience, software development, and electronics. As a senior-year student double majoring in both Computer Engineering and Neuroscience and minoring in Computer Science, I have these necessary backgrounds academically. Furthermore, through UC's cooperative education (CO-OP) program, a 3-year undergraduate research assistantship at MIND Lab, and volunteer-based community outreach activities related to STEM, I also have the practical work and research experience necessary to apply these academic backgrounds to the project.

<u>Advisor (Dr. Rashmi Jha)</u>

Dr. Jha is both my technical advisor for this senior design project and my research advisor at MIND Lab. She has expertise in subjects including neuromorphic devices and computing, emerging logic and memory devices, and emerging nanoelectronic device-based hardware security. She brings a lot of relevant knowledge and experience to the table through years of leading her PhD, masters, and undergrad students in their projects, including ones meant to combine biology and technology, implement bio-inspired computing, and analyze biological signals through software or hardware-based methods.

Project Goals

Goal 1: Create an EEG headset capable of collecting data.

- Conduct literary and market research into available EEG technologies and available headsets.
- Decide on and gather necessary components and materials.
- Build a functioning EEG headset.

Goal 2: Create a labeled dataset (EEG activity is the data, character being presented is the label).

- Design experiment and decide on data type, length, shape, etc.
- Create data collection process and associated programs.
- Follow process to create a labeled dataset.

Goal 3: Create ANNs to analyze the labeled dataset.

- Implement CNN (convolutional neural network) and optimize parameters
- Try various data normalization techniques for input data (note their effects on CNN metrics)
- Find correlations/associations/patterns present in the data

Discussion

Project Concept

Electroencephalography (EEG)

EEG (electroencephalography) maps brain activity by recording the summed electrical signals of different brain regions through non-invasive electrodes placed on the wearer's scalp. EEG headsets are relatively common and cheap compared to other brain imaging equipment, and they have the added benefit of being mobile as the electrodes and EEG board can be contained in the headset.

Artificial Neural Network (ANN)

ANNs (artificial neural networks) are loosely based on biological neural networks. ANNs are usually implemented in software, and they have "neurons" to process signals and "synapses" to give the connections carrying those signals weights. Specifically, CNNs (convolutional neural networks) are shown to be useful in accurately classifying time-series data like EEG recordings.

Combination of EEG and ANN for Project

These two concepts are brought together to create a prototype which can, generally, read in a user's EEG-based brain activity and then to correctly classify that activity. In the specific application chosen for my project, this prototype will be able to present the user with a character (a digit 0-9 or a letter A-J), collect the user's EEG-based brain activity during that time, and process that collected EEG data to 'guess' what character the user saw.

Design Objectives

After some brainstorming sessions, 6 overarching objective categories were found for this project. These categories are ease of use, battery life, comfort, safety, accuracy, and functioning as intended. With these objectives in mind, relevant constraints, functions, and means were developed, and the attribute table in **Figure 1** was created.

Attribute Table				
Characteristic	Objective	Constraint	Function	Means
Be Easy to Use: Headset should be easy to put on and take off	X			
Have Long Battery Life: Power to EEG boards should be long enough to ensure no disruption during data collection process	X	X		
Be Comfortable: Headset should be comfortable enough to not distract wearer (lightweight, cushioned)	X			
Be Easy to Use: Battery for the system should be rechargeable & easily accessible	X			
Be Safe: Headset must be safe for the wearer (i.e. don't electrocute the wearer while checking electrode resistance)	X	X		
Have High Accuracy: Must classify characters beyond chance	X	X	X	
Function as Intended: Headset must classify brainwaves and provide feedback in real time	X		X	
Dry electrodes will be used over wet electrodes				X
OpenBCI Cyton & Daisy EEG boards used for measuring brain activity				X
Properly rated, lightweight, rechargeable LiPo batteries or regular battery holders in combination w/rechargeable AA/AAA batteries will be used				X
Headset and assosiated mounting hardware will be primarily 3D printed				X

Figure 1 - Attribute Table

While the objectives can be viewed as my project's goals, the constraints can be seen as its boundaries. In this attribute table, there are 3 constraints, and each of these constraints is also an objective, so there is overlap between the two categories. The constraints focused on in this attribute table are:

- The battery must have enough power to the headset to avoid disrupting data collection.
- The headset must be safe for the wearer.
- The ANN must classify at an accuracy which is beyond chance.

To figure out which of the overarching objectives needs to be prioritized in this project, a pairwise comparison chart of the overarching objectives is shown in **Figure 2**. The associated weights in each row were calculated using my best judgement based on this project's attribute table, pros and cons of brain activity reading headsets currently on the market, and my own preferences as I will be the main wearer of this headset during the prototyping phase.

Objectives	Easy to Use	Long Battery Life	Comfortable	Safe	High Accuracy	Function as Intended	Score:
Easy to Use	-	0.25	0.5	0	0	0	0.75
Long Battery Life	0.75	-	0.75	0	0	0	1.5
Comfortable	0.5	0.25	-	0	0	0	0.75
Safe	1	1	1	-	1	0.75	4.75
High Accuracy	1	1	1	0	-	0	3
Function as Intended	1	1	1	0.25	1	-	4.25

Objectives	Score:
Safe	4.75
Function as Intended	4.25
High Accuracy	3
Long Battery Life	1.5
Easy to Use	0.75
Comfortable	0.75

Figure 2- Pairwise Comparison Chart [TOP] Detailed Score Breakdown [BOTTOM] Score Summary

The most important objectives are for the product to be safe, for it to function as intended, and for it to have high accuracy. The least important objectives are how comfortable the headset is, the headset's ease of use, and the length of the battery life. This means that as design decisions need to be made, the comfort, ease of use, and battery life aspects of the headset are most likely to get cut out of the final design while the rest of the objectives are integral to the product and will be kept until the end. Now that we know the goals and boundaries of my project, we can move on to identifying the steps necessary for realizing the project.

The necessary devices and steps to get from the expected input (brain activity when user is presented with a character) to the desired output (classification of which character the user saw) were put into the chart shown in **Figure 3.** The top row with purple blocks shows the major steps (ordered from left to right) which will be necessary to complete the project. The green arrows point from each major step to its respective blue block, which shows what is required for that step to be completed.

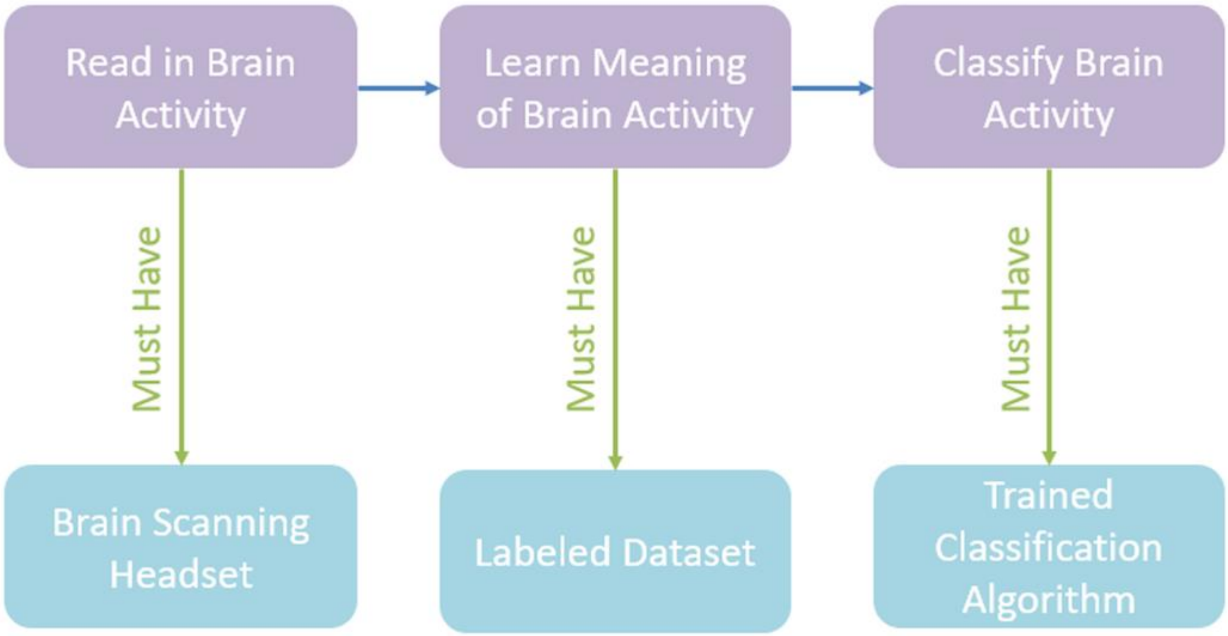


Figure 3 - Required Steps and Components for Reaching Project Goals

To start with, we need to read in the wearer's brain activity. To do that, we must have an EEG headset. Next, we need to learn the meaning of the brain activity we collect. To do that, we must have a labeled dataset. Finally, we must classify the character seen and identified by the user in the recorded brain activity in real time. To do that, we must have a trained classification algorithm. Each of these major steps (which I consider the major goals of this project) will be further detailed in the next section.

Methodology/Technical Approach

Goal 1 (EEG Headset)

Literary and Market Research

Before starting on this project, I first completed some literary research in the hopes of finding signs that my project idea was feasible. The following papers (and video summaries) are examples which detail ANNs classifying a variety of EEG signals gave me hope that this project was possible.

- Convolutional Neural Networks for processing EEG signals (5 minute video about below paper)
    - https://www.youtube.com/watch?v=_z0JIP7pYwE&feature=youtu.be
- On the Classification of SSVEP-Based Dry-EEG Signals via Convolutional Neural Networks
    - https://arxiv.org/pdf/1805.04157.pdf
- A Single-Channel SSVEP-Based BCI Speller Using Deep Learning
    - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8576511
- World's Fastest Brain-Computer Interface: Combining EEG2Code with Deep Learning
    - https://www.biorxiv.org/content/biorxiv/early/2019/02/11/546986.full.pdf
- EEG-Signals Based Cognitive Workload Detection of Vehicle Driver using Deep Learning
    - http://icact.org/upload/2018/0663/20180663_finalpaper.pdf

- A Novel Approach to Emotion Recognition Using Local Subset Feature Selection and Modified Dempster-Shafer Theory
    - https://behavioralandbrainfunctions.biomedcentral.com/articles/10.1186/s12993-018-0149-4

After finishing the literary research, I performed some market research to find the best EEG boards for my purposes. I decided to use the company OpenBCI, because its hardware is opensource, you get access to the raw data, the EEG equipment is research grade, there is an active developer community and support network, and both wet and dry EEG options are offered (more on this later). Also, their products follow the 10/20 placement system (electrode placement for various channel configurations which is agreed upon by the scientific community), so I can use and refer to previous and future EEG studies with ease. For my recordings, I chose OpenBCI's Cyton and Daisy EEG boards. The Cyton board has 8 channels and can record, store, and transfer brain signal information. The Daisy board is an add-on which attaches to the Cyton board to add an extra 8 channels to record from. In total, I will have 16 channels to record from with the Cyton and Daisy combination.

EEG Headset Decisions

After the board selection, I had to choose the electrode type (and based on that, the headset type). Specifically, I had to decide whether to use an electrode cap with wet electrodes, as shown in the left image of **Figure 4**, or a 3D printed headset with dry electrodes, as shown in the right image of **Figure 4**.

Figure 4 - [LEFT] OpenBCI Electrode Cap with Wet Electrodes [RIGHT] OpenBCI 3D Printed Headset with Dry Electrodes

Wet electrodes require electrode gel and (to make things easier) an electrode cap for placement. With wet electrodes, I would get better signal quality due to the electrode gel, however, that signal quality would degrade over time due to the gel drying leading to inconsistencies between recordings or within a single long recording. Also, the setup process is cumbersome, requiring the application of gel on each electrode and the careful placement of each electrode to the scalp (or the cap). This is time consuming, inconvenient, and more difficult to do alone.

With dry electrodes, I would get slightly worse signal quality and more artifacts in my readings, but the readings would be consistent between and within recordings. Furthermore, the headset would be very easy to set up and use, even if putting it on alone, as the electrodes are held in place by the headset frame and can be adjusted to the size and shape of a wearer's head within a few minutes. In future recordings, the wearer simply needs to put the headset on and start recording. For these reasons, I chose the dry electrodes and the 3D printed UltraCortex III headset from OpenBCI.

EEG Headset Construction

The first part of my project included gathering all the materials. I ordered wires, electrodes, screws, custom springs, super glue, batteries, zip-ties, and the like online and had them delivered to my home. I 3D printed the EEG headset using the open-source files for OpenBCI's UltraCortex (UC) III headset. The 3D printed parts included electrode holder components, the headset frame, and the EEG board holder. A photo of the various materials used in the building of the EEG headset is shown in **Figure 5.**

Figure 5 - Headset Build, All Materials

The first parts to be assembled were the electrodes. This process included integrating the AgCl "spikey" (for hair) and "cup" (for skin) electrodes with jumper cables and being placed into the electrode holders. The individual materials used for the electrodes, as well as the assembled electrodes themselves, are shown in **Figure 6.**

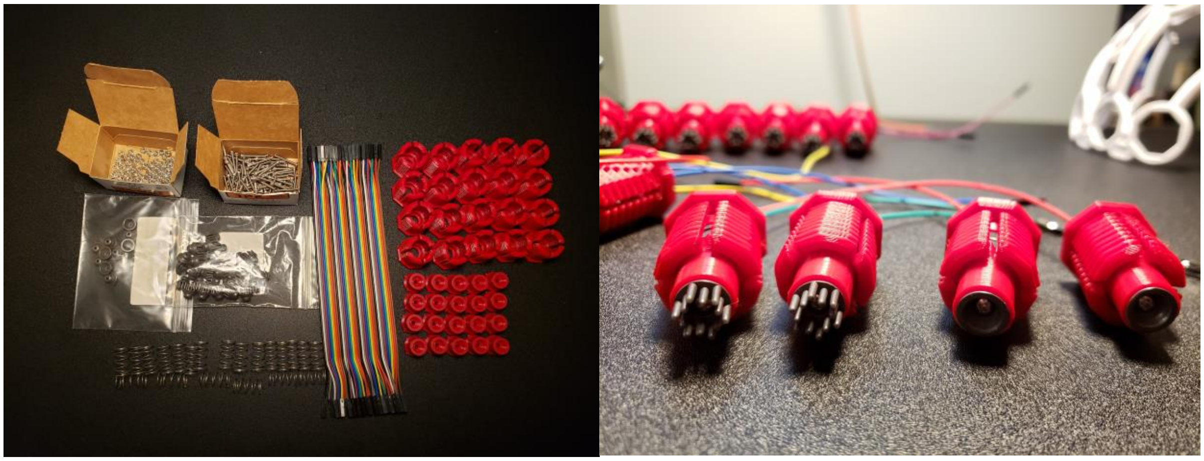


Figure 6 - Headset Build, Electrodes

After the electrodes were completed, I moved on to the EEG headset frame. The white frame itself was made of 2 large parts (the front and the back of the headset) and these two parts were connected by zip-ties. Once the green, threaded, octagonal rings which hold the electrodes in place were glued into the frame, the electrodes were screwed into them. This process is shown in **Figure 7.**

Figure 7 - Headset Build, UltraCortex III 3D Printed Frame

Finally, the green EEG board holder was mounted on the headset, and the Daisy and Cyton boards were screwed into it. Photos of the finished headset and how it sat on my head are included in **Figure 8.**

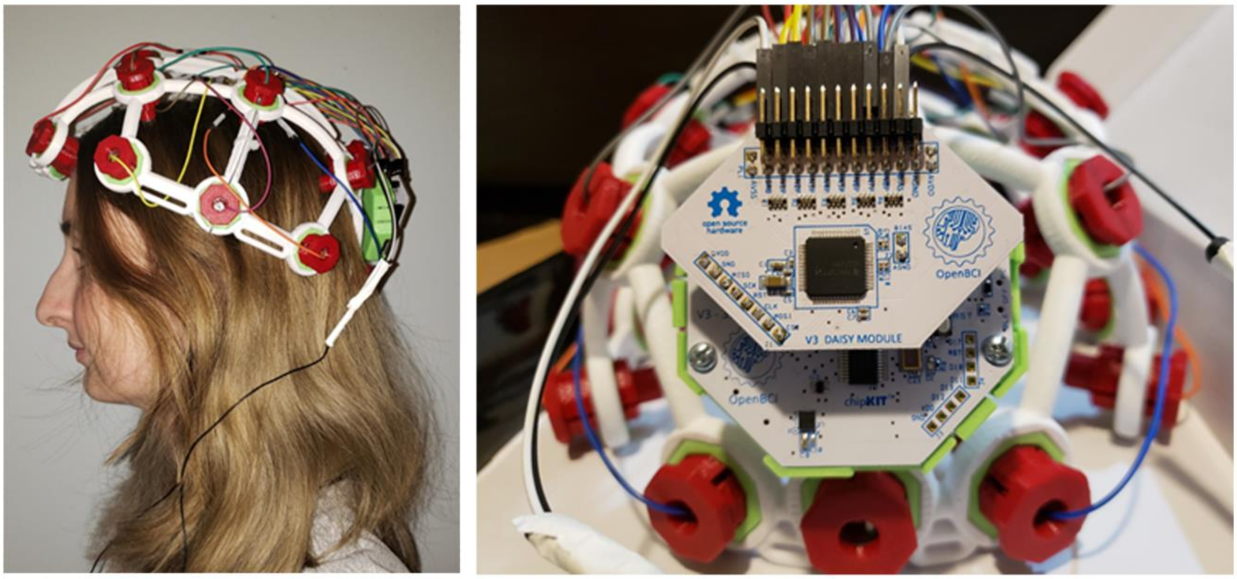


Figure 8 - Headset Build, Finished UC III EEG Headset with Mounted EEG Boards

Goal 2 (Labeled Dataset)

Experiment Design

The first part of the experiment design process was deciding in what way I'd be presenting myself with a stimulus, in this case 20 individual characters (the numbers 0-9 and the letters A-J). A relatively simple but promising solution was to have images containing each character of interest and to then flash those images on a screen while collecting my brain's activity through EEG. The goal would be to record the EEG activity of my brain reacting to an image of a character, specifically of identifying that character. This solution seemed like it would work well, and I moved on to potential constraints for it.

There were several considerations in designing the experiment, and most of these aimed to ensure the integrity of my data. One example is the order in which the numbers and letters would be flashed on the screen – if the order remained the same each session, then I would learn that order over the thousands of sessions required for the data collection process. This learning of the order would mean I'd be expecting specific numbers or letters to show up at different times which would mess up the EEG recordings meant to contain my brain activity of identifying each new character. Because of this consideration, the numbers and letters will be shuffled every session.

Another example of these considerations involves making sure that I don't lower the integrity of my data by cutting corners. One idea for more efficiency and to save time would be to give myself real-time feedback on my current CNN's classification of my brain activity as I am collecting the data. However, this could have disastrous consequences. Instead of recording the EEG data for '7' or 'E', the data collection program could be collecting the EEG data for 'Why is the network messing up again?' or 'Oh good, it's working!' Because of this consideration, I want to give myself as little feedback as possible during the data collection process. I only need to know when the data collection process starts and when it ends.

One final example consideration was looking forward to the future of when I'd be implementing classification. I knew that I wanted a dataset which had an equal (or near equal) number of samples for each character, because if I had certain characters with significantly larger numbers of samples, then the network would just always predict those characters because that would give it its best chance of being correct. Because of this consideration, I made sure to collect an equal number of samples for each character during each data collection session.

Data Collection Process

All data collection and data analysis for Spring 2020 was done in Ubuntu 18.04 (Fall 2019 primarily used Windows 10, but as the application changed, so did some of the tools I used, and a change was made to Ubuntu). The general steps a user would need to follow the data collection process would be:

- Data Collection Prep
    - Launch the OpenBCI GUI, plug in OpenBCI Bluetooth dongle, turn on Cyton EEG board
    - Connect to headset via GUI, start data stream, start LSL stream
    - Place headset on head and ensure adequate channel connectivity
- Data Collection Steps (repeats for each session)
    - User launches the Python-based data collection process
    - Data collection involves 2 rounds: number round (0-9) and letter round (A-J).
    - Each round collects EEG information for 10 characters. For each char:
        - The next character from a shuffled list of all characters is chosen
        - The data collection program connects to the LSL stream from the GUI
        - An image of the character is presented to the user
        - EEG data is recorded from the stream for a little less than 1 second (for 96 records)
        - After 96 records are reached, EEG data stops being recorded
        - The image of the character disappears
        - The program disconnects from the LSL stream
        - EEG recording is saved as a numpy array and labeled via the folder architecture of the save path

Figure 9 shows some screenshots of the "Data Collection Instructions" document which goes into more detail on the process.

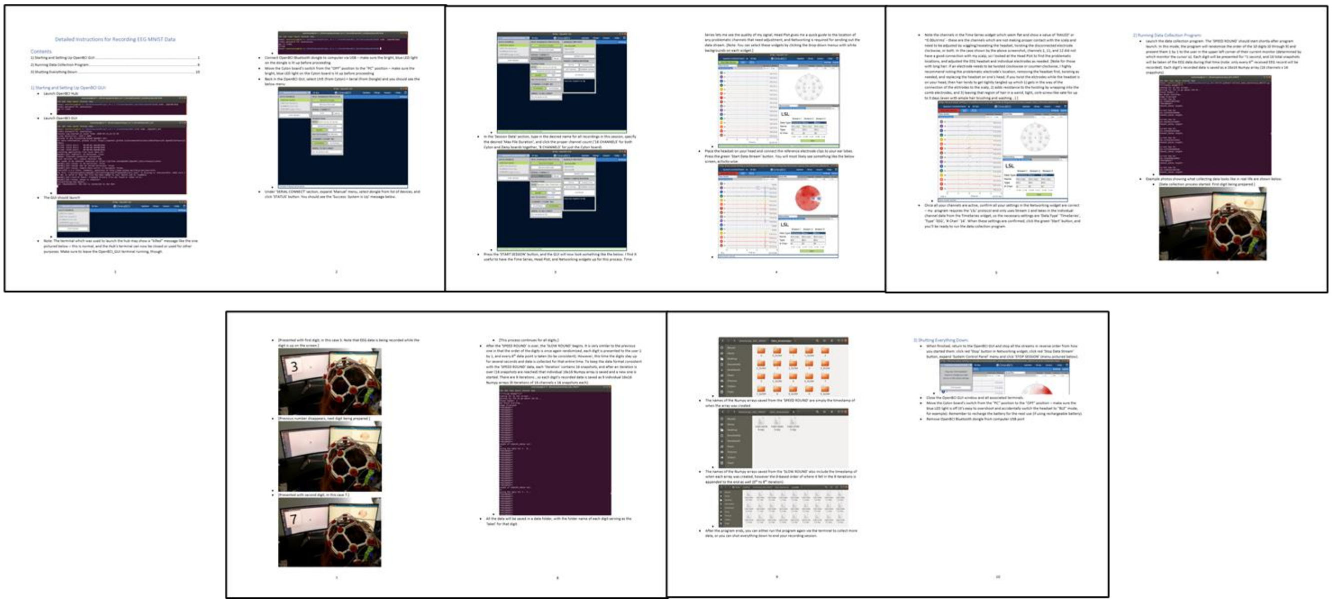


Figure 9 - Screenshots of Data Collection Process Document.

__Collected Dataset__

I have collected 3,160 total samples through the aforementioned data collection process. Figure 10 shows the number of samples per number (0-9) and letter (A-J) used during this project.

- 0: 158 samples
- 1: 158 samples
- 2: 158 samples
- 3: 158 samples
- 4: 158 samples
- 5: 158 samples
- 6: 158 samples
- 7: 158 samples
- 8: 158 samples
- 9: 158 samples

- A: 158 samples
- B: 158 samples
- C: 158 samples
- D: 158 samples
- E: 158 samples
- F: 158 samples
- G: 158 samples
- H: 158 samples
- I: 158 samples
- J: 158 samples

Figure 10 - Table, Number of Samples Per Label (Left = Numbers, Right = Letters)

From the above table, it is clear that each character has the same number of samples to support it (158 samples). This was by design and ensured that the network would not have any incentive in the distribution of sample for guessing the same character at all times.

__Goal 3 (ANN)__

__Data Normalization (and Visualization)__

My EEG headset has 16 channels through which I can read the voltage at different regions of my scalp. During the data collection process for each sample, I get 96 sequential records from each of these 16 channels. This means that I have a 16 by 96 2D array of voltage values. Now, each electrode has a different level of connectedness to my scalp – what I mean by that is some electrodes have a stronger connection than others. However, these readings are all relative to the ground electrodes anyway, and this signal strength changes each time I reseat my headset on my head and the electrodes make contact with my scalp in a slightly different way. To fix these irregularities, I normalized my data. Normalizing across the board didn't seem to help much, so I normalized the values per sample per channel. That means that for each individual sample (each 96 records of looking at a specific character) I would normalize each individual channel in that sample (I would take each of the 16 channels and normalize its values to be from 0 to 1). This seemed to help a lot more.

To get a better sense of the data I was working with, I decided to visualize it in an image, so I exported each of my 3,160 samples into a '.png' file. Below I will show a few examples of these visualizations. The y-axis is the channels (channels 0 to 15), and the x-axis is the records (records 0-95). The color shows the normalized, relative voltage value and ranges from yellow to dark purple. The yellow parts have the highest values, the green parts contain values in the middle, and the dark purple (almost black) parts include the very low values. **Figure 11** shows two different but typical samples of the number '0.'

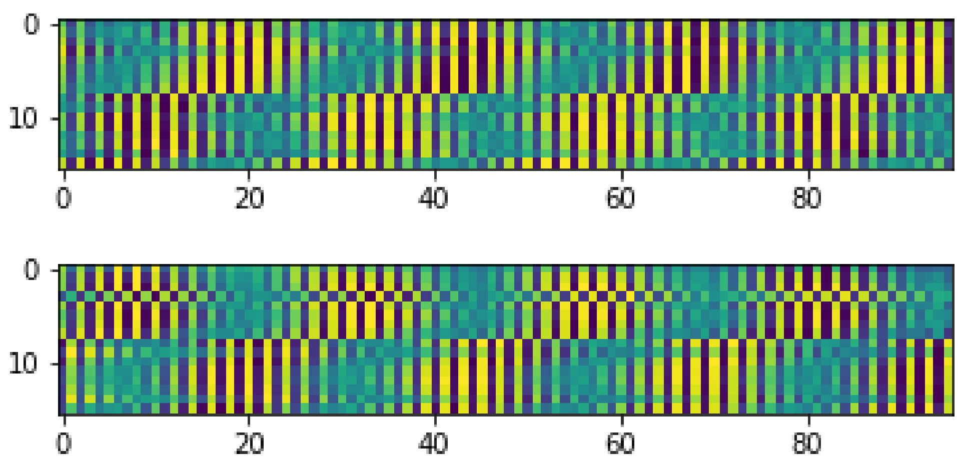


Figure 11 - Visualizations of 2 Typical Samples of '0'

There are several characteristics that can be visually noted about the above two samples. First off, there are green 'waves' that periodically show up in the visualization. Another characteristic is that the reading frequently alternates between close to the maximum (highest, yellow) value of each channel for each sample and close to the minimum (lowest, purple) value of each. These low and high points for each channel seem to line up with the low and high point for each other channel within one board. Channels 0-7 (the first 8 channels) are located on the Cyton board, which is the one that communicates with the PC to stream the EEG data via Bluetooth. Channels 8-15 (the next 8 channels) are located on the Daisy board, which is a physical add-on connected to the Cyton board. Looking at the green 'waves', while the channels seem in sync internally (within each board), they seem out of sync externally (between each board). In other words, it looks like the Cyton channel data and the Daisy channel data is out of sync or out of phase. Just by looking at how clearly separated the readings are between channels 1-8 and 9-16, I'd say the difference in phase would probably be due to the boards rather than any biological phenomenon - this could easily be checked by swapped a channel from the Cyton and Daisy boards and collecting some samples to compare to. Some potential issues which could cause this shift include the physical hardware connection causing a delay, a setting in the GUI software, or the way the Cyton board firmware receives Daisy data and combines it to be sent out. This can be looked into further in future work, but I don't believe it should cause problems for my project as this phenomenon is consistent and the network should be able to handle it thanks to that consistency.

While the above two samples were pretty typical for a visualization of any of my EEG recordings from this project, not all samples look the same. Some of them seem very distorted and out of sync with each other. An example of one of these noisy or atypical samples of the number '0' is shown in **Figure 12**.

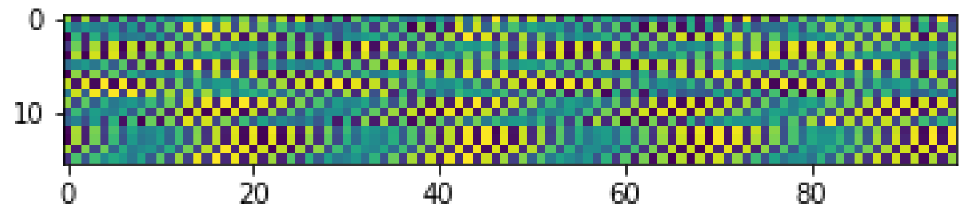


Figure 12 - Visualization of a Noisy/Atypical Sample of '0'

There are many potential causes for this type of a sample. There could be an interruption in the Bluetooth stream for example, or I could have introduced noise through grinding my teeth or moving my head. Alternatively, this may not be 'noise' alone – it could at least in part be caused by some biological process in my brain. Looking into and trying to replicate phenomenon like these can be looked into further in future work.

ANN Creation

My ANN went through several iterations, but overall it has been a relatively simple CNN, like one you would see used for basic image recognition tasks like classifying the MNIST dataset, which served as the inspiration for this project, or a cats vs dogs image dataset. I had many problems early on with getting the network to be compatible with the data I collected and with getting the network to learn. There were many changes initiated which allowed the network to learn and at least overfit on the collected EEG data, and some of these changes are discussed below.

Before we get into the improvements, though, we need to know what my starting point was when I initially created and tried to test my networks. I had a little under 1,000 samples and I had a relatively small batch size. My data was also initially completely unnormalized, as some papers had shown they could classify on the raw EEG data itself, but that proved not possible in my case, so I normalized it across the whole dataset. This was not effective as each sample, and more importantly each channel in each sample, had a different level of connectivity to my scalp and therefore a different voltage range that it was recording in. This problem made me wonder whether the change in each channel was more important than simply the value of each channel. This led to the idea of manipulating my data so that the change in the channel was the input to the network and seeing if it would improve my network's metrics – but more on this later.

Another major aspect to note at my starting point was that my initial EEG data was made up of numpy arrays which were 16 by 16 square 'images' instead of 16 by 96 rectangular 'images'. This 16 by 16 square was the initial shape of my data because I thought there would be some benefit from classifying a square image over a rectangular one. However, since 16 records go by very quickly during the data collection process, I didn't have the time to see each character come up. Because of that lack of time, I decided to sample once every x records or the median of the previous x records, so I used trial and error to choose an x which left the character up long enough for me to recognize it but not too long that I would start thinking about it or looking around and wondering what I'm supposed to do. 6 seemed to serve well as my x, so I sampled once every 6 records. While the time the character was up worked well, after a short while I realized I was losing too much resolution with the downsampling, so I stopped discarding the other 5 records and ended up with the full 96 records per sample. Now that we understand a little about my starting point, let’s go on to the changes I made.

The following changes yielded improved results in the network:

- Collecting all 96 records per sample instead of downsampling to 16
- Collecting more EEG data for the dataset
- Increasing the batch size to the entirety of the training dataset
- Normalizing the data per channel per sample

Interestingly, manipulating my newly normalized data to show the change in each channel instead of simply the normalized voltage values experienced in each channel did not seem to improve the network metrics, as shown in the next section on classification results.

Classification Results

Overall, my network did not learn above chance. It could overfit, it could learn the training data it was presented with, but it could not learn the trends in that data and would then have poor accuracy when presented with never-before-seen data. **Figure 13** shows an example of this overfitting. The left image in this figure shows the confusion matrix for classifying the digits 0-9 in a network which was trained and tested on the same data – training here lasted for 200 epochs. The right image in this figure shows the confusion matrix for classifying those same digits in a network which was trained on training data for 1000 epochs and was tested on separate testing data.

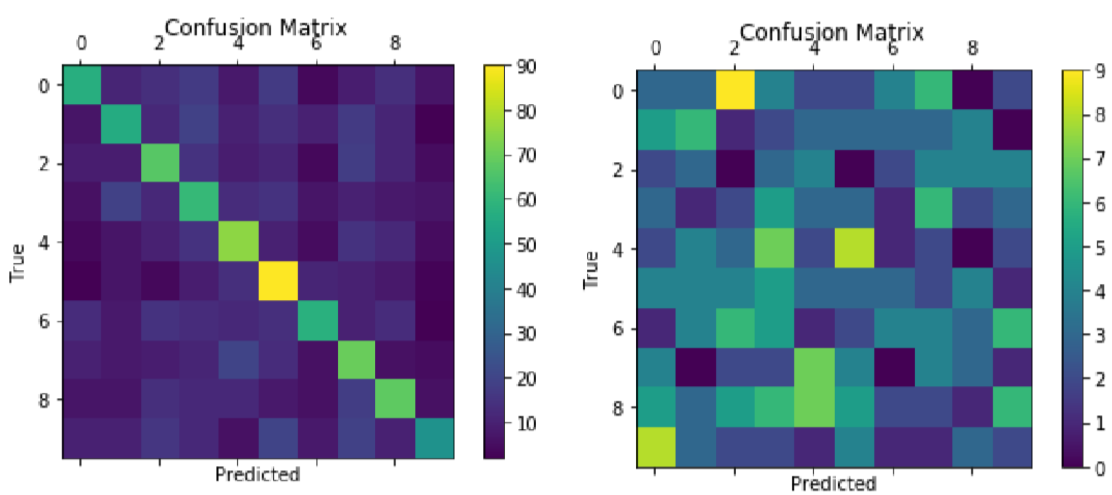


Figure 13 - Confusion Matrices of Digits. [LEFT] Train & Test on All Data, 200 Epochs [RIGHT] Train on Train Data, Test on Test Data, 1000 Epochs

While the left image shows that most of the true values (labels) of the data and the values predicted by the network matched up, as we would want to see for high accuracy, it's important to remember that this network was trained and tested on the exact same data. If instead you test on different data, as in the right image, then even as you increase the number of epochs, the network predicts values seemingly at random. This is a great example of overfitting and learning the training data instead of the trends in the data.

A similar phenomenon can be seen in **Figure 14**. The left image in this figure shows the confusion matrix for classifying the letters

in a network which was trained and tested on the same data – training here lasted for 200 epochs. The right image in this figure shows the confusion matrix for classifying those same digits in a network which was trained on training data for 1000 epochs and was tested on separate testing data.

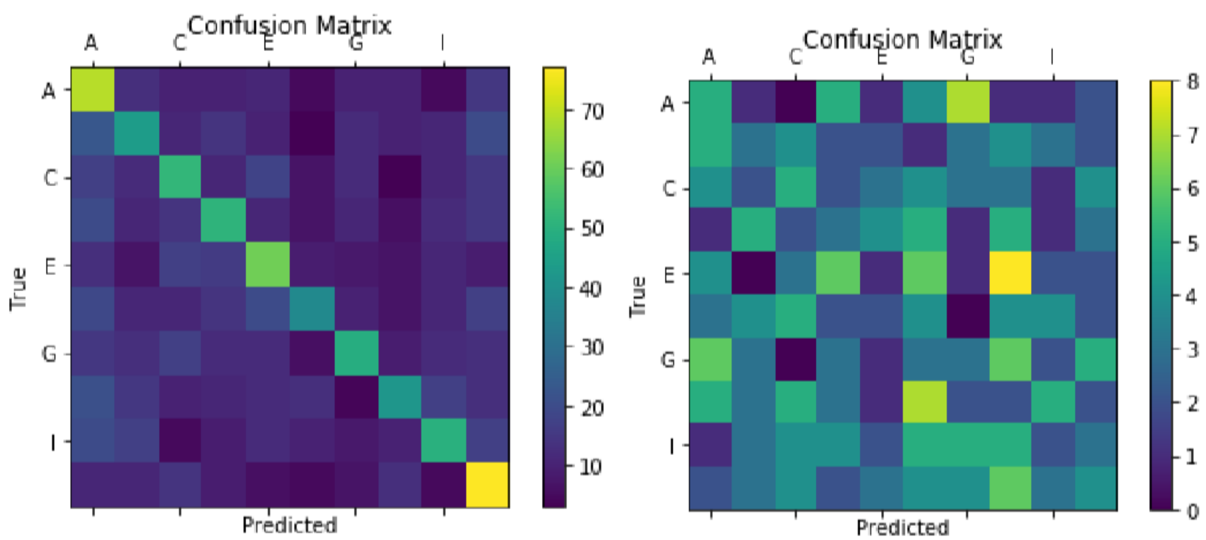


Figure 14 - Confusion Matrices of Letters. [LEFT] Train & Test on All Data, 200 Epochs [RIGHT] Train on Train Data, Test on Test Data, 1000 Epochs

Again, the left image has that nice diagonal clearly visible but the right image seems to be nothing but noise – it's yet another example of overfitting.

When I put all 20 types of characters into the training and testing datasets for the network, something unexpected happened. While in earlier attempts, I saw what seemed like above-chance classification between letters and numbers (letters being made up of all the data for letters A-J and numbers being made up of all the data for digits 0-9), the more data I got and the more changes I made to the network, the less the network for this specific application seemed to learn. I couldn't even get the network to overfit as I did in the above two figures. One explanation could be that there simply isn't enough data for the network to find the similarities between the 10 different numbers and the 10 different letters which unite them as numbers and letters, respectively. It could also easily be that the network is simply too simple to pick out that information from the EEG data. There are many steps which could be taken to further refine what reason is most likely and to attempt to improve the classification, but I have unfortunately run out of time and must stop here for this project.

Finally, I'd like to discuss the idea I had previously about feeding in the change in the EEG data instead of simply the normalized voltage data to the network. The idea behind this data manipulation was that it could be that the change in the electrical activity was more indicative of the phenomenon taking place than the raw voltage values themselves. However, when I implemented this change, I did not see any major improvements. **Figure 15** shows two confusion matrices. Both the left and right images in this figure show confusion matrices for classifying the letters 'A' and 'D' in a network which was trained on training data for 200 epochs and was tested on separate testing data. The difference is that the left image's dataset was simply the raw voltage values which were normalized per channel per sample, while the right image's dataset was the normalized change in the value of each sample's channel.

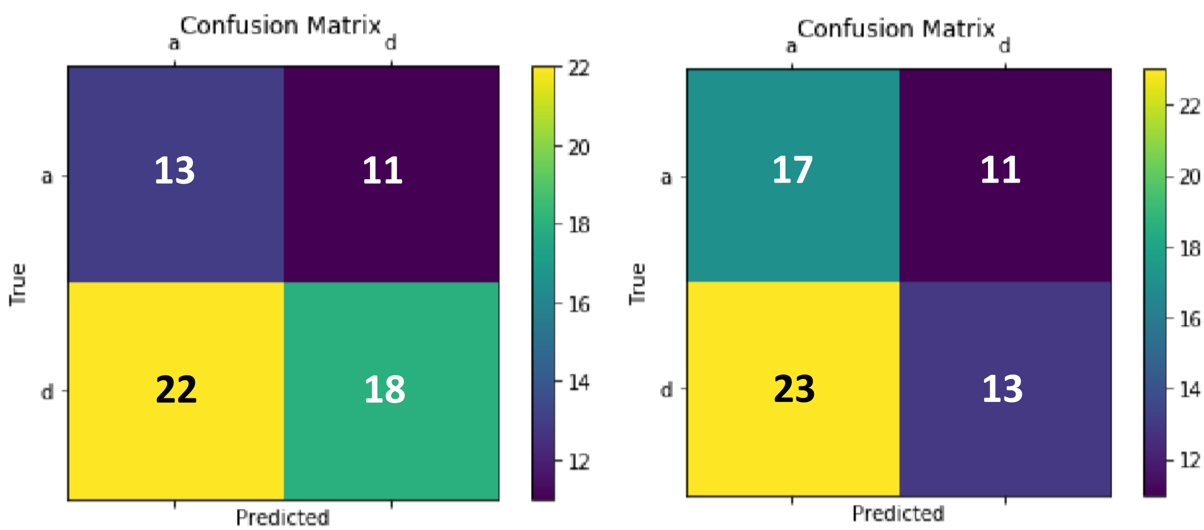


Figure 15 - Confusion Matrices of 'A' and 'D'. [LEFT] Normalized Raw Values, Train on Train Data, Test on Test Data, 200 Epochs [RIGHT] Normalized Change in Values, Train on Train Data, Test on Test Data, 200 Epochs

In the left image, the accuracy was 48.4%, while in the right image, the accuracy was 46.9%. The accuracy alone shows no improvement, but while that change in accuracy is very small, it seems from the confusion matrices that the main difference between the two is that the right one predicted 'A' correctly more often while the left one predicted 'D' correctly more often. However, I believe this change is simply because of the test data distribution of this particular example and does not reflect any major change in the network.

First off, please note that while the testing data in these cases was not evenly distributed, the training data used to set the weights in the networks were. In both cases, the network predicted 'A' more often than 'D' even though the test data (the true labels) had more 'D' samples than 'A'. Even though the samples of the right image's test data are more evenly distributed than that of the left (28 true 'A' and 36 true 'D' for right, 24 true 'A' and 40 true 'D' for left), its predicted values are much more skewed than that of the left (40 predicted 'A' and 24 predicted 'D' for right, 35 predicted 'A' and 29 predicted 'D' for left). It almost seems worse than chance.

Budget

Along with the materials required for the EEG headset, there are also materials required for building that headset. **Figure 16** shows a summary of the cost of my project, including the cost of all materials together (including borrowed and purchased materials), the amount of money borrowed components are worth, and the total of all purchased items.

Total of All Items Needed:	**$**	**1,476.23**
Total of Borrowed Items:	$	957.73
Total of Purchased Items:	$	518.50

Figure 16 - Summarized Budget

The total cost for this project is $1,476.23. However, the most expensive component, the EEG boards ($957.73), was borrowed from the neuroscience department, so the actual total of items which needed to be purchased is $518.50. The BOM (Bill of Materials) which breaks down this cost by component and lists each component's purpose, description, cost, and URL is shown in **Figure 17**. Components with a green background have been bought and are now owned, while components with a yellow background have been borrowed for the duration of the project.

Component Name	Quantity	Unit Price	Shipping,Tax	Total	Planned Usage	Product Description	Product Link
Dry EEG Comb Electrodes [30-pack]	1	$ 29.99	$ 7.74	$ 37.73	EEG electrodes to be used for headset (these are the longer 5mm deep comb electrodes for scalp sections with hair)	The prongs of this dry EEG electrode end in blunt tips for long-term comfort and wearability. The extended (5mm) prongs accommodate longer hair while enabling excellent signal quality	https://shop.openbci.com/products/5-mm-spike-electrode-pack-of-30?variant=8120433606670
Ear Clip Electrodes	2	$ 19.95	$ 3.79	$ 43.69	EEG electrodes to be used for headset	This unit consists of two EEG electrode bodies in a plastic ear clip with a lead terminating in a MedSafe™ DIN connector.	https://www.fri-fl-shop.com/product/td-430-silver-disc-electrode-ear-clip/?attribute_pa_length=48in
Dry/Wet EEG Cup Electrode [10-pack]	1	$ 7.00	$ -	$ 7.00	EEG electrodes to be used for headset (these are for scalp sections without hair)	This disposable/reusable Wet EEG Electrode can also be used for other parameters such as EMG, ECG or EOG. It is used with the TDE-207XX Lead Wire.	https://www.fri-fl-shop.com/product/tde-201/?attribute_pa_quantity=10
Black 4" zip ties [100-pack] & White 4" zip ties [100-pack]	1	$ 5.69	$ -	$ 5.69	used for securing loose wires on the headset	The Self-locking 4 inch nylon cables ties are good for home organizing, and the fastening cable tie can secure a banner that be displayed	https://www.amazon.com/Mudder-Nylon-Cable-Black-Pieces/dp/B017B6CMQK/ref=sr_1_4?keywords=zip+ties+tiny&qid=1570479558&sr=8-4
Male-Male, Male-Female, and Female-Female Jumper Cables [40-pack of each]	1	$ 6.59	$ -	$ 6.59	Cables to be used for connecting Cyton & Daisy boards to the EEG electrodes on the headset	Including 1X7.9in 40 pin male to female jumper wires,1X7.9in 40 pin male to male jumper wires, 1X7.9in 40 pin female to female jumper wires	https://www.amazon.com/Eiechip-Dupont-Jumper-Breadboard-Multicolored/dp/B07RN4WN2N/ref=sr_1_2_sspa?keywords=male+to+female+cables&qid=1570387781&sr=8-2-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzVElxSU1ET0daTlMzJmVuY3J5cHRlZElkPUEwMTczMTIwMlAxSjVXNlVPUVk2RiZlbmNyeXB0ZWRBZElkPUEwODc2ODAxQVU3RkxDTEI1MjdKJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==
Cyton + Daisy Biosensing Boards (16-Channels)	1	$ 949.99	$ 7.74	$957.73	EEG Boards to be mounted on headset and used for reading EEG signals from electrodes	The OpenBCI Cyton Board and OpenBCI Daisy Module (which plugs into the OpenBCI Cyton Board) can be used to sample up to 16 channels of brain activity (EEG), muscle activity (EMG), and heart activity (ECG).	https://shop.openbci.com/collections/frontpage/products/cyton-daisy-biosensing-boards-16-channel?variant=38959256526
Lithium Ion Polymore Battery 3,7V @500mAh	1	$ 9.53	$ -	$ 9.53	battery pack to power the Cyton & Daisy boards	Lithium ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light and powerful. The output ranges from 4.2V when completely charged to 3.7V. This battery has a capacity of 500mAh for a total of about 1.9 Wh	https://www.amazon.com/ADAFRUIT-INDUSTRIES-1578-Lithium-Polymer/dp/B00L0W61VO/ref=sr_1_1_sspa?keywords=Lithium+Ion+Polymer+Battery+-+3.7v+500mAh&qid=1570480190&sr=8-1-spons&psc=1&smid=AM0JQO74J587C&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzVllPNTVFSzVCTUQwJmVuY3J5cHRlZElkPUEwODM2NTQ0UFVJNFo0UzBXTkM5JmVuY3J5cHRlZEFkSWQ9QTA4MDgzMTcyWllWSlBFRktCNE9MJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==
Adafruit Micro Lipo - USB Battery Charger	1	$ 7.70	$ -	$ 7.70	charger for the above battery pack	Oh so adorable, this is the tiniest little lipo charger, so handy you can keep it any project box! Its also easy to use. Simply plug in the gold plated contacts into any USB port and a 3.7V/4.2V lithium polymer or lithium ion rechargeable battery into the JST plug on the other end.	https://www.amazon.com/Adafruit-Micro-Lipo-Charger-ADA1304/dp/B00OKJFEQ2/ref=cm_cr_arp_d_product_top?ie=UTF8
SanDisk Extreme Pro 64GB micro SD card	1	$ 19.50	$ 1.84	$ 21.34	SD card for collecting the data without being tethered to the computer or using bluetooth	Up to 170MB/s read and 90MB/s write speeds for fast shooting and transfers	https://www.amazon.com/SanDisk-Extreme-MicroSDXC-UHS-I-Adapter/dp/B07G3GMRYF
Inland Premium PLA+ 3D printer filament 1,75mm [1kg]	2	$ 18.99	$ -	$ 37.98	Printer filament to be used for 3D printing OpenBCI UltraCortex headset, altered custom headset, and EEG board and embedded system housings (and associated bio-feedback device housing)	PLA+ is several times tougher than the normal PLA on the market; it also produces Smoother finished printouts with no cracking or brittle problem.	https://www.microcenter.com/product/504222/premium-175mm-white-pla-3d-printer-filament---1kg-spool-(22-lbs)
LocTite Super Glue	1	$ 6.75	$ -	$ 6.75	Super Gluing the 3D printed headset together	Dries transparent for invisible repairs Sets in seconds with no clamping required Super strength formula	https://www.amazon.com/Loctite-Super-Professional-Bottle-1405419/dp/B00JN1E4TK/ref=sr_1_2_sspa?keywords=loctite+super+glue&qid=1570831072&sr=8-2-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUFMQ1ZHUjcxRVJZNFAmZW5jcnlwdGVkSWQ9QTA4MDA3MzUxMzRIT1pLVTROWVpNJmVuY3J5cHRlZEFkSWQ9QTA3MTQwNTEyUkJBV1c3WVpRSUJIJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==
Compression Springs "Weak" U-4	21	$ 2.12	$ 10.57	$ 55.09	for use with electrodes inside the electrode holder	Regular compression spring in stainless steel with 5.50 total coils and closed ends. This spring features an O.D. of 0.420in, an I.D. of .376in, a spring rate of 1.3	https://prod.centuryspring.com/catalog/?page=product&id=U-4CS&_ga=2.173196003.1305963661.1570481625-437807374.1570481625
Compression Springs "Strong" S-845	5	$ 3.33	$ -	$ 16.65	for use with electrodes inside the electrode holder	Regular compression spring in stainless steel with 4.33 total coils and closed/ground ends. This spring features an O.D. of 0.420in, an I.D. of .350in, a spring rate of 14.	https://prod.centuryspring.com/catalog/?page=product&id=S-845CS&_ga=2.173196003.1305963661.1570481625-437807374.1570481625
Stainless Steel Pan Head Phillips Screw [100-pack]	1	$ 4.27	$ 8.07	$ 12.34	for use with electrodes inside the electrode holder	18-8 stainless steel pan head screws have good chemical resistance and may be mildly magnetic. Length is measured from under the head.	https://www.mcmaster.com/91772a084
Combination Slotted/Phillips Rounded Head Screws [100-pack]	1	$ 1.52	$ -	$ 1.52	for use with electrodes inside the electrode holder		https://www.mcmaster.com/90077a106
Stainless Steel Hex [100-pack]	1	$ 3.11	$ -	$ 3.11	for use with electrodes inside the electrode holder	These nuts have good chemical resistance and may be mildly magnetic.	https://www.mcmaster.com/91841a003
Header Pin Connector Kit	1	$ 12.83	$ 0.90	$ 13.73	for use in soldering header pins to bluetooth receiver of Cyton board for firmware update circuit construction	Package Includes: Female / Male Housing Connector, 2.54mm pin connector, 40 Pin 2.54mm Pitch Single / Right Angle Row Pin Headers, Rainbow Color Flat Ribbon IDC Cable.	https://www.amazon.com/gp/product/B01G0I0ZZK/ref=ppx_od_dt_b_asin_image_s00?ie=UTF8&psc=1
NVIDIA Jetson Nano Developer Kit	1	$ 99.00	$ 7.91	$106.91	Primary microcontroler responsible for running the trained algorithm which classifies the user's state of focus in real time and sends signals out to the biofeedback device	NVIDIA Jetson Nano Developer Kit is a small, powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. All in an easy-to-use platform that runs in as little as 5 watts.	https://www.microcenter.com/product/606566/nvidia-jetson-nano-developer-kit
MicroSD Card 32 GB UHS1	1	$ 13.99	$ -	$ 13.99	microSD card for the NVIDIA Jetson Nano	Record and play Full HD video. With ultra-fast read & write speeds up to 95MB/s & 20MB/s	https://www.microcenter.com/product/476233/samsung-32gb-evo-microsdhc-class-10-uhs-1-flash-memory-card-with-adapter
5 inch Raspberry Pi Screen	1	$ 36.99	$ 7.55	$ 44.54	display for Jetson Nano, mounted in arm gauntlet (forearm)	5 inch HDMI display with 800*480 LCD screen	https://www.amazon.com/Elecrow-800x480-Interface-Supports-Raspberry/dp/B013JECYF2/ref=sr_1_7?keywords=hdmi+display+screen&qid=1574479403&sr=8-7
Mini Keyboard with Touchpad	1	$ 16.95	$ -	$ 16.95	for navigating jetson nano/troubleshooting/launching apps, Mounted in arm gauntlet (forearm), may be removable for ease of use	3 in 1 Multifunction: Wireless Keyboard + Touchpad + Mouse.	https://www.amazon.com/Rii-Wireless-Keyboard-Touchpad-Controller/dp/B006WAJ1IE/ref=sr_1_10?keywords=mini+folding+keyboard+trackpad&qid=1574481362&sr=8-10
Jeston Nano Clear Case + Fan	1	$ 11.99	$ -	$ 11.99	a case for the jetson nano so its components aren't as exposed when mounted in the gauntlet	good-looking, dust-proof; clever open design; easy to access the peripheral connectors; Reserve a small fan hole, Comes with Fan	https://www.amazon.com/MakerFocus-Acrylic-Developer-3-0-5-8V-Enclosure/dp/B07VL2BG29/ref=sr_1_13?keywords=jetson%2Bnano%2Bcase&qid=1574477412&sr=8-13&th=1
Power Bank (26750mAh, 5V/2x2.4A)	1	$ 29.99	$ -	$ 29.99	powers gauntlet (jetson nano, screen, keyboard+trackpad), mounted in arm gauntlet (bicep)	DC 5V/2.4A charger, 26750mAh, 2 output ports	https://www.amazon.com/dp/B00MQSMEEE/ref=psdc_7073960011_t2_B00FDK2G2C
Power Adapters - Barrel Jack	1	$ 7.69	$ -	$ 7.69	to connect high amperage output port of power bank to barrel jack input of jetson nano	Connectors: USB Type-A Male to 5.5x2.1mm Barrel Male, with 4 connectors: 2.5x0.7mm, 3.5x1.35mm, 4.0x1.7mm, 5.5x2.5mm	https://www.amazon.com/5-5x2-1mm-Positive-Charger-connectors-Compatible/dp/B07QCT65SP/ref=sr_1_1_sspa?crid=3VPKHEIGV8FI&keywords=usb+to+barrel+jack&qid=1574478931&sprefix=usb+to+barrel%2Caps%2C152&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyT0pJNFVEWEVSTkRNJmVuY3J5cHRlZElkPUEwNjk4OTU4MzJSUkU3QUFQQjdWUSZlbmNyeXB0ZWRBZElkPUEwMTA1MTc0MjY1T0pMOVAyQzZENyZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=

Figure 17 – Detailed Bill of Materials (BOM)

Timeline

Fall Semester

In the fall semester, my main responsibilities were completing initial literary research, gathering necessary materials and building my EEG headset, completing initial data collection, and starting to analyze the collected data. A Gantt chart showing what my Fall 2019 timeline looked like at the end of November is shown in **Figure 18**.

WBS	Tasks	Start	End	% Complete	Duration (Days)	Hours Spent	23 - Sep - 19	30 - Sep - 19	07 - Oct - 19	14 - Oct - 19	21 - Oct - 19	28 - Oct - 19	04 - Nov - 19	11 - Nov - 19	18 - Nov - 19	25 - Nov - 19	02 - Dec - 19	09 - Dec - 19
1	**Literary Reasearch**	9/22/19	10/26/19	100%	35	11.71												
1.1	EEG Studies (especially related to focus) from the past [passiv ebackground research Part 1]	9/22/19	10/6/19	100%	15	9.46												
1.2	Neuromorphic Component, Softw are and/or Hardw are [passiv e background research Part 1]	10/7/19	10/26/19	100%	20	2.25												
2	**Gathering Materials & Building Headset**	10/7/19	11/8/19	97%	33	26.76												
2.1	Gather EEG boards, w ires, electrodes	10/7/19	10/8/19	100%	2	0.58												
2.2	3D print EEG headset pieces	10/9/19	10/14/19	100%	6	0.50												
2.3	Put together headset	10/12/19	10/14/19	100%	3	5.75												
2.4	Update firmw are on boards and get all softw are installed	10/14/19	11/4/19	100%	22	16.76												
2.5	Design data labeling method for recording	10/27/19	11/3/19	80%	8	0.00												
2.6	Practice In-Class Recordings, Adjust Headset	11/4/19	11/8/19	100%	5	3.17												
3	**Initial Data Collection**	11/11/19	11/22/19	100%	12	7.36												
3.1	Record First Week	11/11/19	11/15/19	100%	5	4.92												
3.2	Record Second Week	11/18/19	11/22/19	100%	5	2.44												
4	**Analyze Data**	11/25/19	12/8/19	0%	14	0.00												
4.1	Record Holiday Week	11/25/19	11/27/19	0%	3	0.00												
4.2	Algorithm Dev elopment/ Data Analy sis	11/25/19	12/8/19	0%	14	0.00												
4.2.1	Create Initial Analy sis Algorithm	11/25/19	12/6/19	0%	12	0.00												
4.2.2	Find What States are Classifiable	12/7/19	12/8/19	0%	2	0.00												

Figure 18 - Gantt Chart, Fall 2019

One thing that should become evident form looking at this Gantt chart is that the project application was different from what it is now. Specifically, the application I was working toward in Fall 2019 was concentration assistance. However, while the data collected during this time did not carry over to the final application, many of the other completed tasks, like literary research, building the headset, and brainstorming data collection protocols did. With this disclaimer out of the way, we can look at a further breakdown of the hours spent on this project in Fall 2019 to see how my hours were distributed in the early days of this project.

Figure 19 shows a summary of the hours I had spent on this project until 11/24/2019. The left chart shows the number of hours worked for each individual week (there had been 10 weeks at that point). The right pie chart shows the category breakdown of hours spent on this project (with 103.13 hours being the total for those 10 weeks).

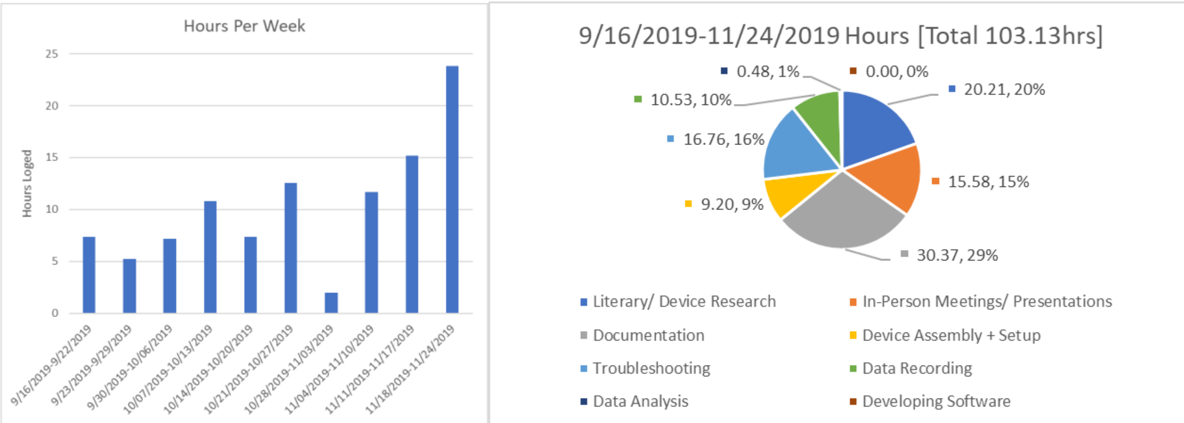


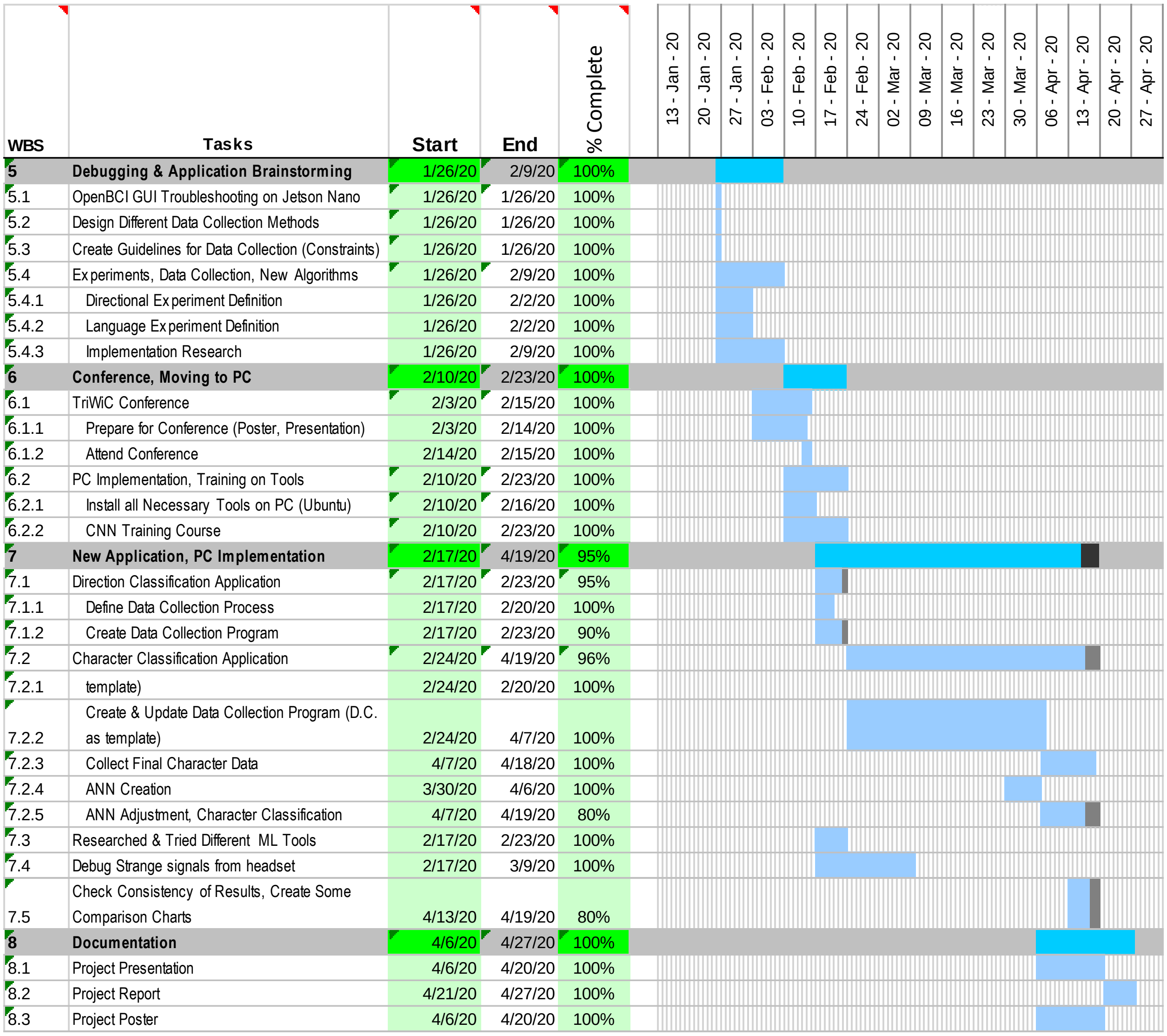
Figure 19 - [LEFT] Hours Spent by Week [RIGHT] Total Hours Spent by Category

Overall, the hours per week trend was increasing – as I completed parts of my project and became able to do more activities with it, I spent more time on it. The pie chart shows that my hours were skewed toward documentation, literary research, meetings, and troubleshooting and skewed away from the data collection and analysis which this project revolves around. However, there is a reasonable explanation for this trend: 46.17 hours, or 45% of the time at that point, was made up of one-time events or startup tasks which are not expected to recur. These events are Device Assembly + Setup (9.20 hours or 9% of total time), EEG board firmware troubleshooting (16.76 hours or 16% of total time) and Literary/ Device Research (20.21 hours or 20% of total time). While at this point, only 11.01 hours or 11% of the time has been used to collect data or analyze it, that number is expected to increase greatly as there are no longer any obstacles preventing EEG activity recording.

Another notable trend is that documentation of the project, which includes making presentations, time keeping, and writing documents, took up approximately 29% of my time, and this trend is expected to continue as planned overhead. While this overhead is unfortunate in the sense that I could be using that time to collect and analyze more data, it is still necessary and useful for myself and for future engineers who may wish to learn from this project.

<u>Spring Semester</u>

In the spring semester, my application changed from concentration improvement to direction classification. Shortly thereafter, the direction classification experiment and data collection process was altered to become the character recognition application which this paper describes. My main responsibilities for the spring 2020 semester were redefining the experiment and tools necessary to complete it, creating new protocols and programs for data collection, collecting new data using those protocols, creating an ANN to analyze that data, and finally documenting the entire project with the presentation, report, and poster. A Gantt chart showing what my Spring 2020 timeline looks like now is shown in **<u>Figure 20</u>**.



WBS	Tasks	Start	End	% Complete
5	**Debugging & Application Brainstorming**	1/26/20	2/9/20	100%
5.1	OpenBCI GUI Troubleshooting on Jetson Nano	1/26/20	1/26/20	100%
5.2	Design Different Data Collection Methods	1/26/20	1/26/20	100%
5.3	Create Guidelines for Data Collection (Constraints)	1/26/20	1/26/20	100%
5.4	Experiments, Data Collection, New Algorithms	1/26/20	2/9/20	100%
5.4.1	Directional Experiment Definition	1/26/20	2/2/20	100%
5.4.2	Language Experiment Definition	1/26/20	2/2/20	100%
5.4.3	Implementation Research	1/26/20	2/9/20	100%
6	**Conference, Moving to PC**	2/10/20	2/23/20	100%
6.1	TriWiC Conference	2/3/20	2/15/20	100%
6.1.1	Prepare for Conference (Poster, Presentation)	2/3/20	2/14/20	100%
6.1.2	Attend Conference	2/14/20	2/15/20	100%
6.2	PC Implementation, Training on Tools	2/10/20	2/23/20	100%
6.2.1	Install all Necessary Tools on PC (Ubuntu)	2/10/20	2/16/20	100%
6.2.2	CNN Training Course	2/10/20	2/23/20	100%
7	**New Application, PC Implementation**	2/17/20	4/19/20	95%
7.1	Direction Classification Application	2/17/20	2/23/20	95%
7.1.1	Define Data Collection Process	2/17/20	2/20/20	100%
7.1.2	Create Data Collection Program	2/17/20	2/23/20	90%
7.2	Character Classification Application	2/24/20	4/19/20	96%
7.2.1	template)	2/24/20	2/20/20	100%
7.2.2	Create & Update Data Collection Program (D.C. as template)	2/24/20	4/7/20	100%
7.2.3	Collect Final Character Data	4/7/20	4/18/20	100%
7.2.4	ANN Creation	3/30/20	4/6/20	100%
7.2.5	ANN Adjustment, Character Classification	4/7/20	4/19/20	80%
7.3	Researched & Tried Different ML Tools	2/17/20	2/23/20	100%
7.4	Debug Strange signals from headset	2/17/20	3/9/20	100%
7.5	Check Consistency of Results, Create Some Comparison Charts	4/13/20	4/19/20	80%
8	**Documentation**	4/6/20	4/27/20	100%
8.1	Project Presentation	4/6/20	4/20/20	100%
8.2	Project Report	4/21/20	4/27/20	100%
8.3	Project Poster	4/6/20	4/20/20	100%

Figure 20 - Gantt Chart, Spring 2020

Problems Encountered/Analysis of Problems Solved

Hardware/Software Issues

The EEG boards I borrowed had never been used before, and their firmware was several years out of date. While initially it seemed like a relatively quick and simple fix, the process of updating the boards became a surprisingly effective time sink. There were many problems encountered while trying to update the firmware, which included having to download and install no longer available and conflicting tools, conflicting instructions and missing update information, and having to build a hardware circuit to allow for the update in the first place, and this entire process took several weeks instead of the few hours I had planned for it.

Dry EEG Readings with Long Hair Issues

Once I got the firmware updated, I could connect the board to my PC and start recording and viewing the EEG activity from all 16 channels. However, when I connected, there would be 5 to 6 channels that would show no activity. I had just built my headset so it was untested, and I had just barely updated the board so the correctness of that update was questionable. I wasn't sure if it was the headset, the board, or something else that was causing those channels to not show up. After a few days, I placed the headset on the head of somebody with short hair, and all the channels worked perfectly – the conclusion was that it was my long hair that was causing the signal quality issues. I ended up finding a way to make everything work by pre-adjusting the headset to my head size and shape while it was not on my head so the electrodes wouldn't get wrapped in my hair, and then placing it on my head and giving the electrodes minute, final adjustments based on the signal quality shown from the readings. It works very quickly and easily now.

Application Change Issues

This project went through several redefinitions. Specifically, the goal went from being concentration improvement to direction identification to character recognition. These changes in project application and goals cost me valuable time and caused my early efforts to become inapplicable to the final project. However, hindsight is 20-20 and this was the path I needed to take to get to where I am now. Even though not everything carried over from one stage to the next, progress was made each step of the way. The concentration improvement phase of the project saw the building of the EEG headset and my familiarization with the OpenBCI GUI and initial brainstorming on data collection processes. The direction identification phase of the project saw the development of a better data collection process and the coding of the data collection software which would be used as a template for the final application. In the final application, the character (digit or letter) recognition phase, I used those accomplishments from the previous parts and built upon them to make the project described in this paper.

Pandemic-Related Issues

While initially I thought that the pandemic, the shelter in place orders, and UC moving online would not have had much of an impact on my project, I've realized that its impact on me has impacted the project. I had only seen the positives of this pandemic. For example, being the only student member of this team, I wasn't separated from any teammates and did not have to deal with trying to find ways to meet with others or exchange project materials during a pandemic. I also had the benefit of building my project from scratch and having all the devices on hand at my home – I therefore would still have unlimited access to the hardware (EEG headset) even while my laboratory was closed. However, I failed to account for the fact that the rest of my life would be affected by the pandemic and would require some adjusting to. Academically, for example, there were a lot of changes which served as a time sink. Each of my classes took a different approach to going online, many syllabuses and class plans got changed, and each professor seemed to have their own favorite platform and software suites they wanted to implement for their individual classes. Overall, the impact these changes had on me was that I had much less time than I would have wanted to work on any specific thing, including this project. I simply could not ramp up my work hours for it as easily as I had in the fall.

Other Issues

I have a very busy schedule, probably unusually so for a senior, and while I spent a lot of time on this project, I do wonder where it could have gone if I had a teammate or two putting in that same amount of time. Looking back from where I am now, I realize that I would have probably benefited from having a teammate to share the work with and bounce ideas off of. While having a teammate may have caused its own set of problems once distance learning started and campus closed due to the COVID19 pandemic, I believe the pros of having another knowledgeable, hardworking student working on this project with me would have far outweighed those potential cons.

Future Recommendations/ Future Work

The following tasks are good starting points for continuing this work:

- Understanding, replicating, and explaining the collected data visualizations
    - i.e. explaining phase difference, replicating and explaining atypical visualization
- Getting the neural network to learn the trends instead of simply overfitting
    - Keep changing network parameters
    - Try more neural network architectures
    - Collect more digit and letter data
        - Potentially change the way data is normalized
        - Potentially change the shape of the data (collect for a longer time?)
    - Extract more features from the data

Conclusion

Acknowledgements

- This research project was funded by the following awards from the National Science Foundation:
    - CCF-1718428
    - ECCS-1556294

References

- OpenBCI Documentation
    - https://docs.openbci.com/docs/Welcome.html
- UltraCortex III Documentation
    - https://docs.openbci.com/docs/04AddOns/01-Headwear/MarkIII
- Convolutional Neural Networks for processing EEG signals (5 minute video about below paper)
    - https://www.youtube.com/watch?v=_z0JIP7pYwE&feature=youtu.be
- On the Classification of SSVEP-Based Dry-EEG Signals via Convolutional Neural Networks
    - https://arxiv.org/pdf/1805.04157.pdf
- A Single-Channel SSVEP-Based BCI Speller Using Deep Learning
    - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8576511
- World's Fastest Brain-Computer Interface: Combining EEG2Code with Deep Learning
    - https://www.biorxiv.org/content/biorxiv/early/2019/02/11/546986.full.pdf
- EEG-Signals Based Cognitive Workload Detection of Vehicle Driver using Deep Learning
    - http://icact.org/upload/2018/0663/20180663_finalpaper.pdf
- A Novel Approach to Emotion Recognition Using Local Subset Feature Selection and Modified Dempster-Shafer Theory
    - https://behavioralandbrainfunctions.biomedcentral.com/articles/10.1186/s12993-018-0149-4