MindGames

Project Breakdown

Brandon Knotek (bk11@hood.edu)
Walid Muhammad (wm5@hood.edu)
Jack Wilder (jmw38@hood.edu)

Our Goal

To explore the viability of neurofeedback as a means for controlling real-time interactive experiences

Our Goal

To explore the viability of neurofeedback as a means for controlling real-time interactive experiences

In other words, play a game just by thinking commands; no mouse and keyboard required!

Why Bother?

- **Accessibility** - Physical disabilities may limit use of traditional input methods
- **Communication** - Certain conditions make movement and speech impossible (Locked In Syndrome)

So How Does It Work?

1. Brain activity is recorded and sent to a computer using special hardware
2. The computer processes the activity and translates it to commands
3. The commands are sent to the target software and executed

Unraveling the Brain

- All brain activity is electrical
- Electroencephalography (EEG) uses electrodes placed on the scalp to measure this activity

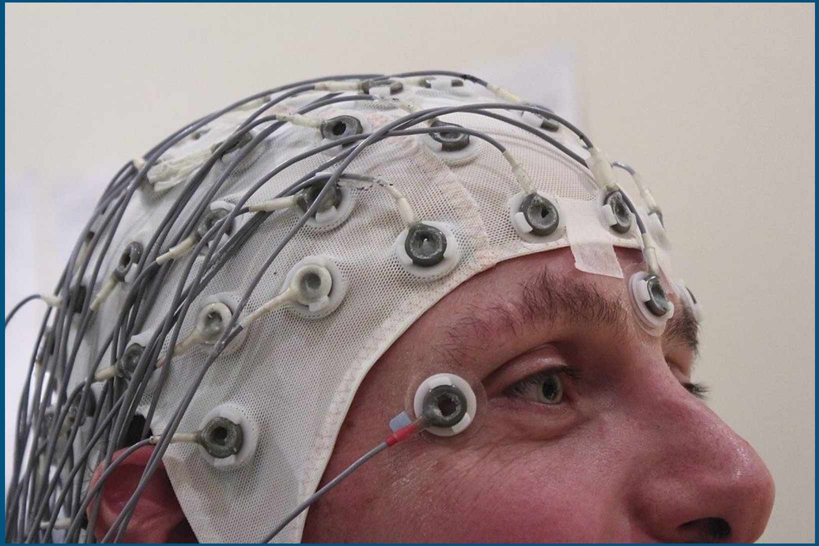

Figure 1 - An example of EEG hardware

Unraveling the Brain

- Different thoughts, actions, and mental states produce unique electrical patterns

- These patterns are different for any given person

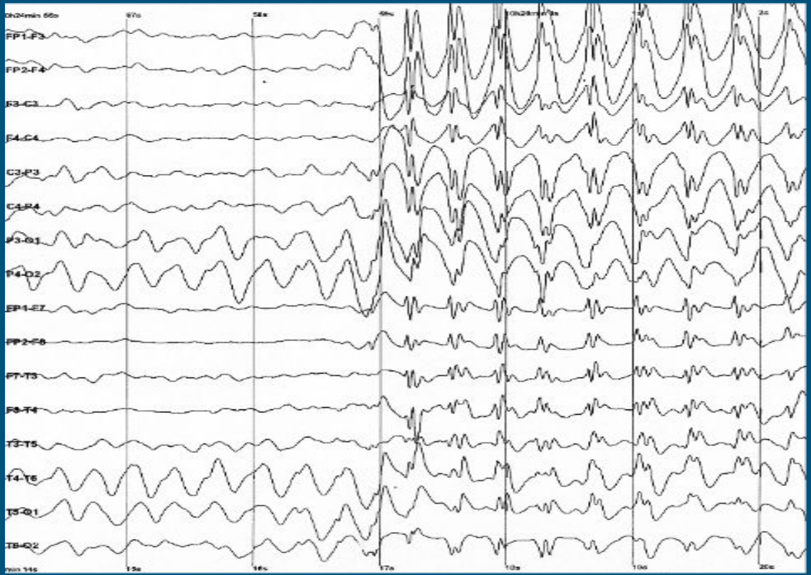

Figure 2 - Sample EEG output

Building a Neural Fingerprint

- The computer must be trained to recognize an individual's unique patterns

- Using machine learning (ML), the computer is taught to associate labeled patterns with commands

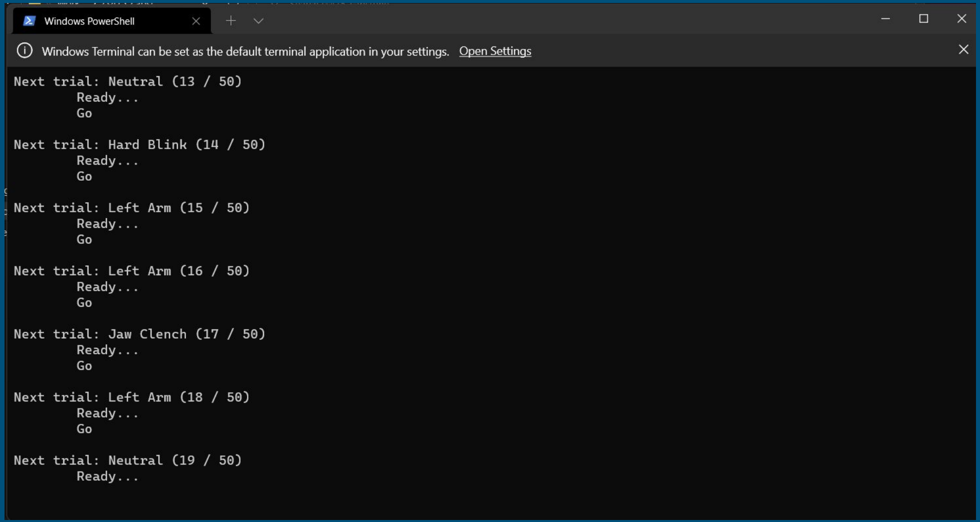

Figure 3 - Our calibration module collecting training data

Command Classification

- With enough examples, the ML model is able to classify subsequent patterns on its own

- Using probability, it determines the command most likely to correspond to a given pattern

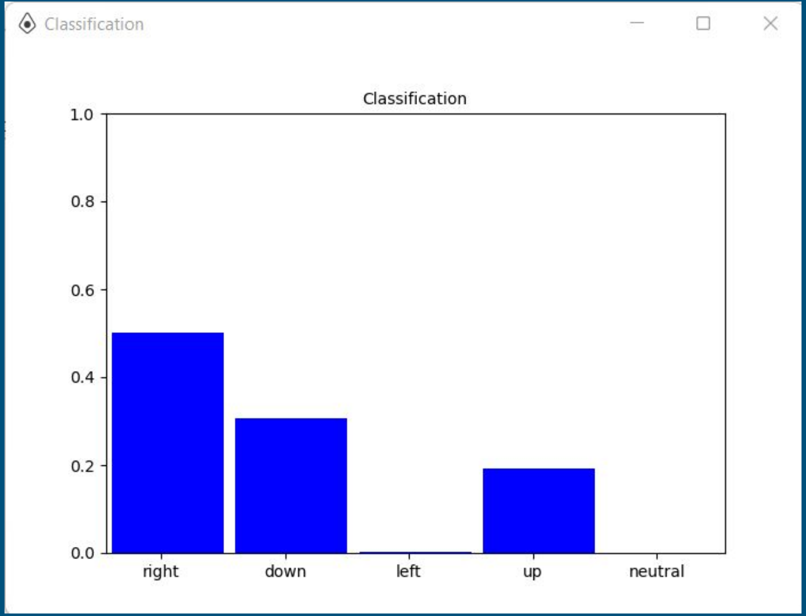

Figure 4 - A visualization of the ML model's classification output

Playing the Game

- The most likely command is sent to the game
- The command is executed and Pac Man moves in the desired direction

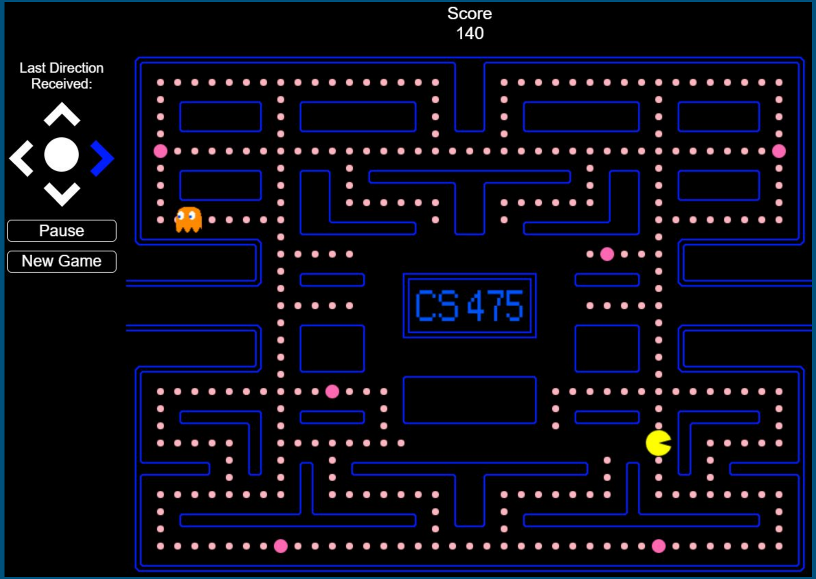

Figure 5 - Pac Man in action

System Overview

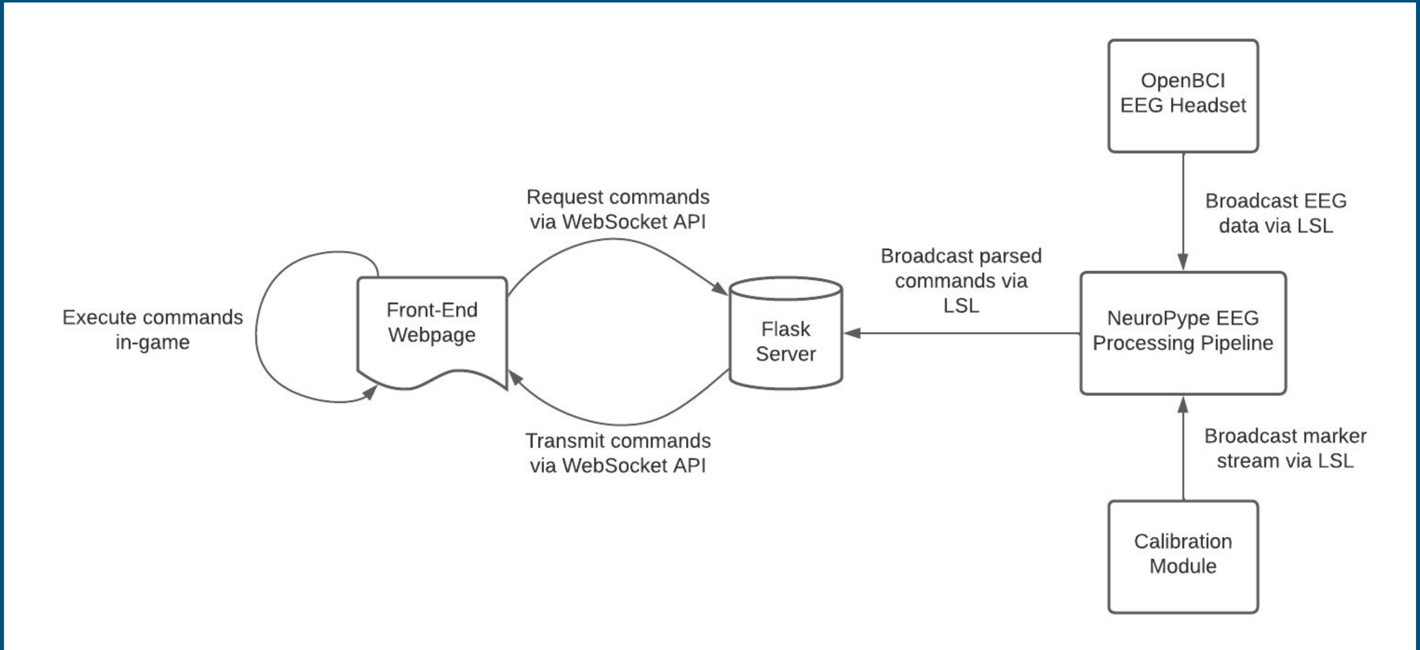

Figure 6 - A diagram of the MindGames system architecture

System Overview - Communication

- **Lab Streaming Layer (LSL)** - Synchronization and transmission of real-time continuous measurements
    - EEG data and training labels

- **WebSocket** - Continuous communication between a webpage and a server
    - Game commands

System Overview - Processing Pipeline

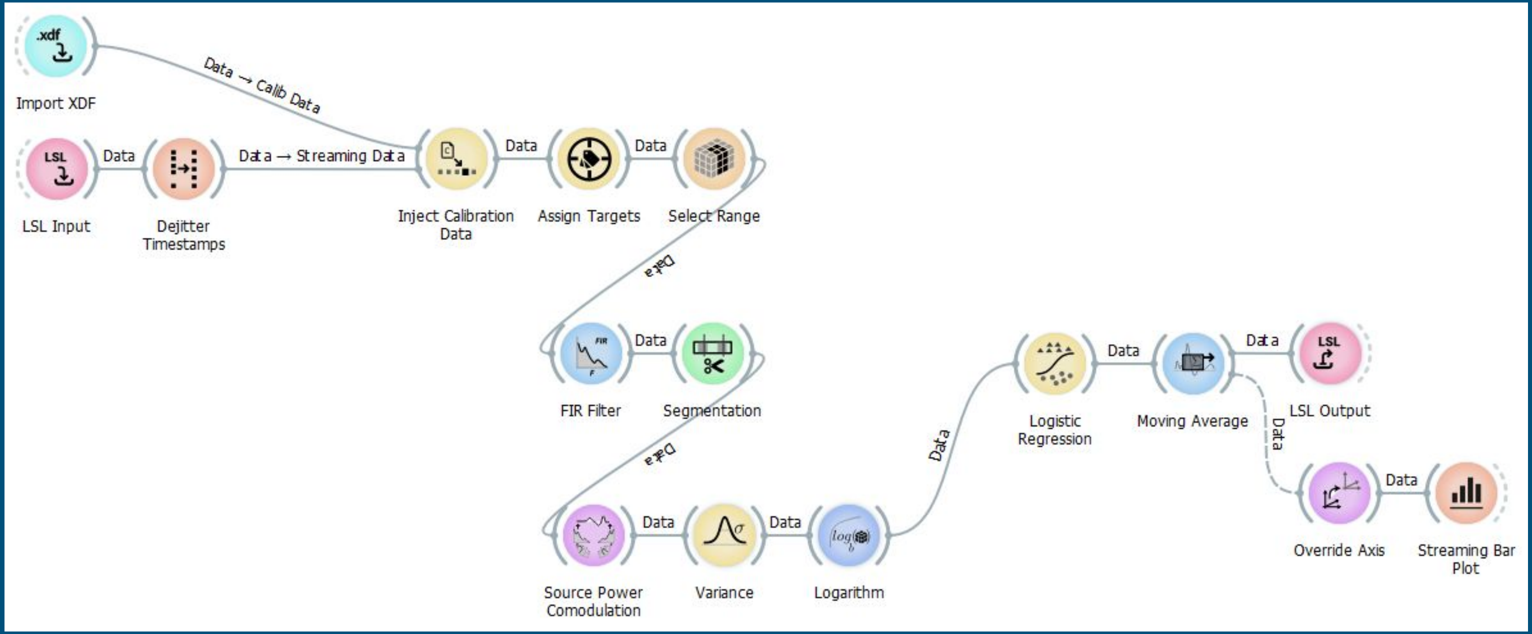

Figure 7 - The MindGames EEG processing pipeline

Processing Pipeline - Data Acquisition

- Get EEG data and ML training labels into the pipeline
- Assign training labels to commands

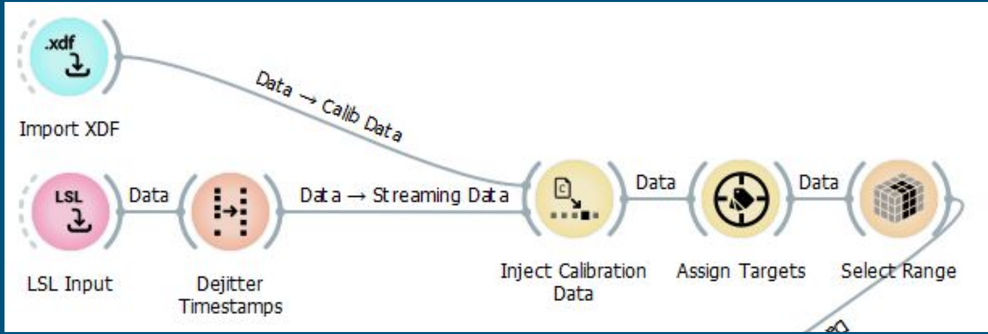

Figure 8 - EEG pipeline data acquisition component

Processing Pipeline - Preprocessing

- Filter out electrical noise
- Designate length of data “chunks” to associate with labels

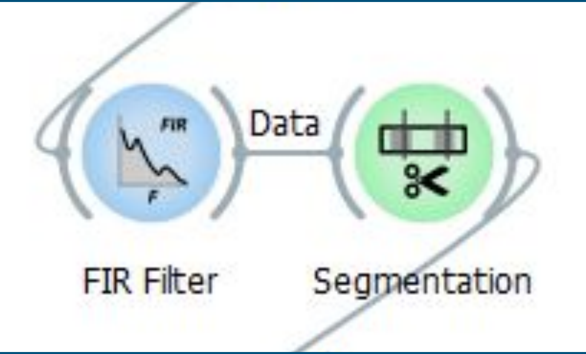

Figure 9 - EEG pipeline preprocessing component

Processing Pipeline - Feature Extraction

- Extract EEG signal components that exhibit the most variance between commands

- Reduce the volume of data to make ML classification more efficient

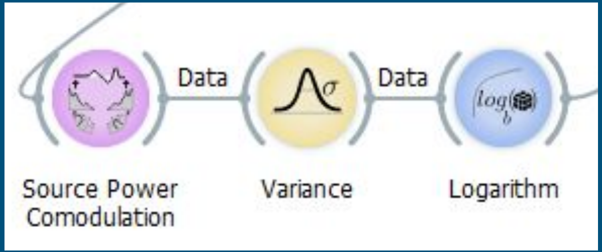

Figure 10 - EEG pipeline feature extraction component

Processing Pipeline - Classification

- Use extracted features to determine command probabilities

- Maintain a moving average of probabilities to reduce noise

- Output probabilities for further use

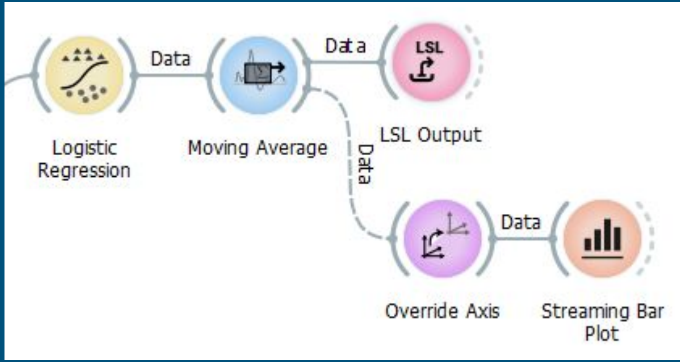

Figure 11 - EEG pipeline classification component

Challenges - Hardware

- EEG data collected using OpenBCI Ultracortex headset
- Difficult to secure, small movements cause it to shift position
- Electrodes move with the headset, generating spikes and invalidating training data

Figure 12 - Ultracortex headset, profile view

Challenges - Hardware

- Limited to 8 electrodes (professional setups may use up to 128)
- Electrodes are not secured to the scalp
- RESULT: Low quality data (and not enough of it)

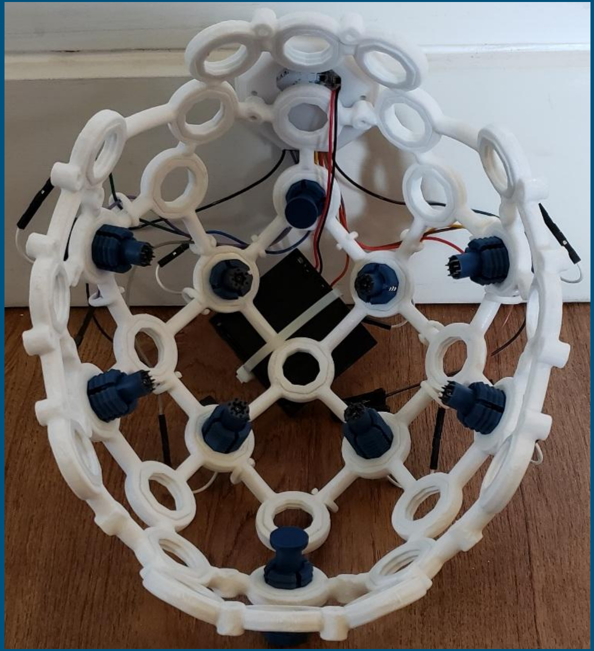

Figure 13 - Ultracortex headset, bottom view

Challenges - Software

- ML training takes 8 minutes, results become useless as electrodes shift with headset
- Noise and irrelevant spikes (e.g., blinking, swallowing) must be filtered out without affecting relevant signals
- Everything must happen with minimal latency for the game to feel responsive

Next Steps

- Fine-tune our noise reduction and ML classification techniques
- Devise a system to secure the headset in place
- Transition from classifying movements to classifying imagined movements (motor imagery)

Questions?