MindGames

CAPSTONE PROJECT PRESENTATION

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

Our Goal

Neurofeedback to control real-time interactive experiences

Our Goal

Neurofeedback to control real-time interactive experiences

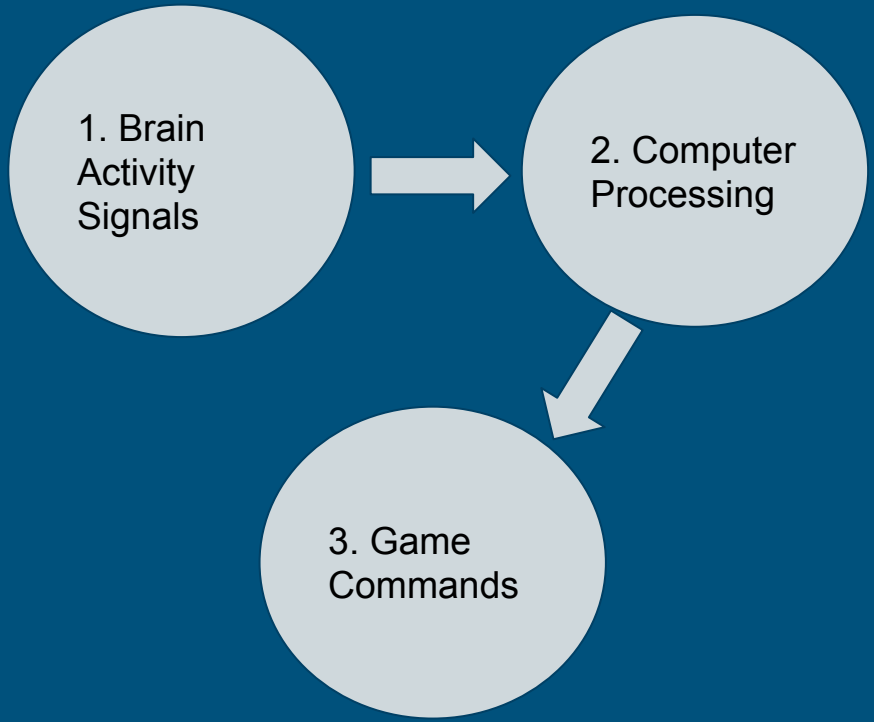
Play Pac-Man just by **thinking** commands

No mouse, No keyboard, No voice

Project Motivation

- **Accessibility**
  - Physical disabilities
- **Communication**
  - Locked In Syndrome
- **Immersive Experiences**
  - Games and simulations

System Requirements



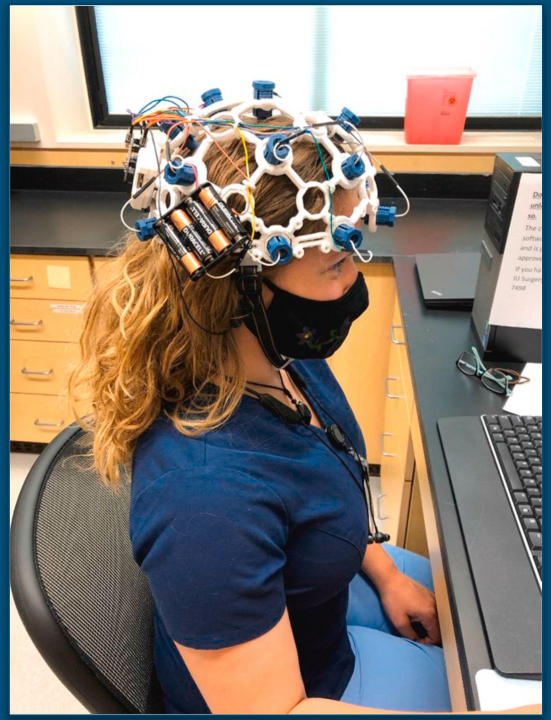


Fig. 1 - Using an OpenBCI Ultracortex headset to interact with a computer

Unraveling the Brain

- All **brain activity = electrical**
- *Electroencephalography* (EEG) uses electrodes to measure activity
- Different **thoughts**, **actions**, and **mental states** produce unique electrical patterns
- Patterns differ between individuals

Fig. 2 - An example of EEG hardware

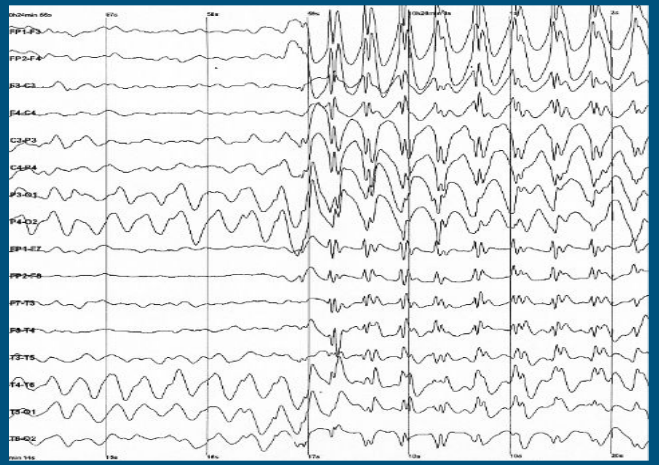


Fig. 3 - Sample EEG output

Building a Neural Fingerprint

- Computer training:
  - Analyze an individual's unique patterns
- Machine Learning (ML):
  - Computer associates labeled patterns with commands

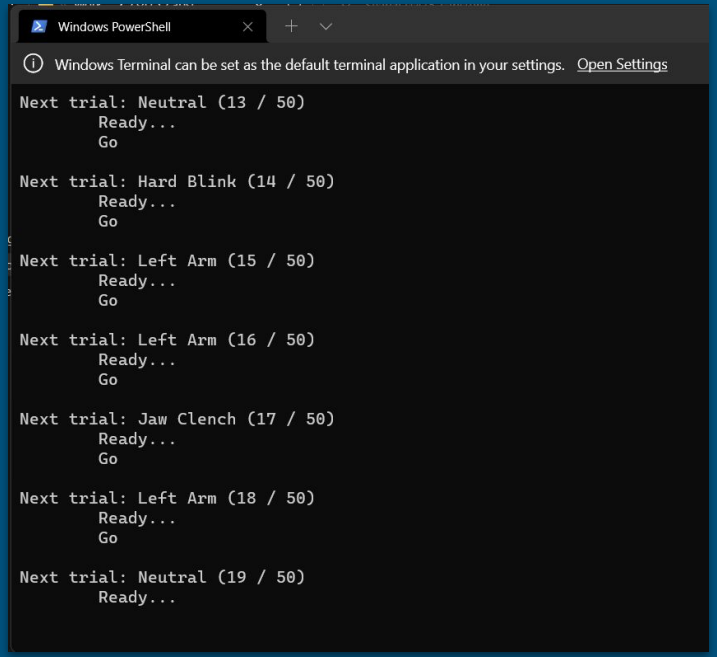


Fig. 4 - Calibration module collecting training data

Command Classification

- Enough examples for ML model to classify patterns on its own
- Probabilistic:
  - Determines **most likely command** given pattern

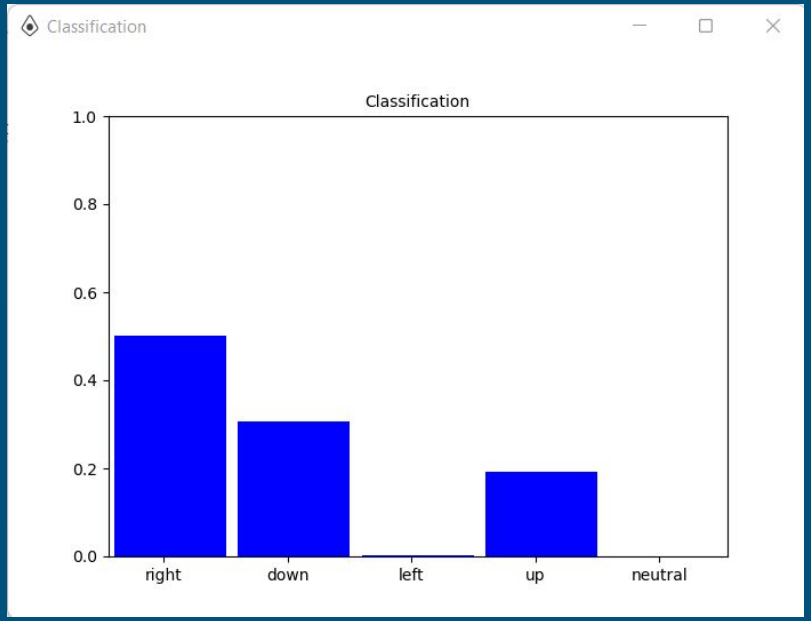


Fig. 5 - Visualization of model's classification output

System Overview

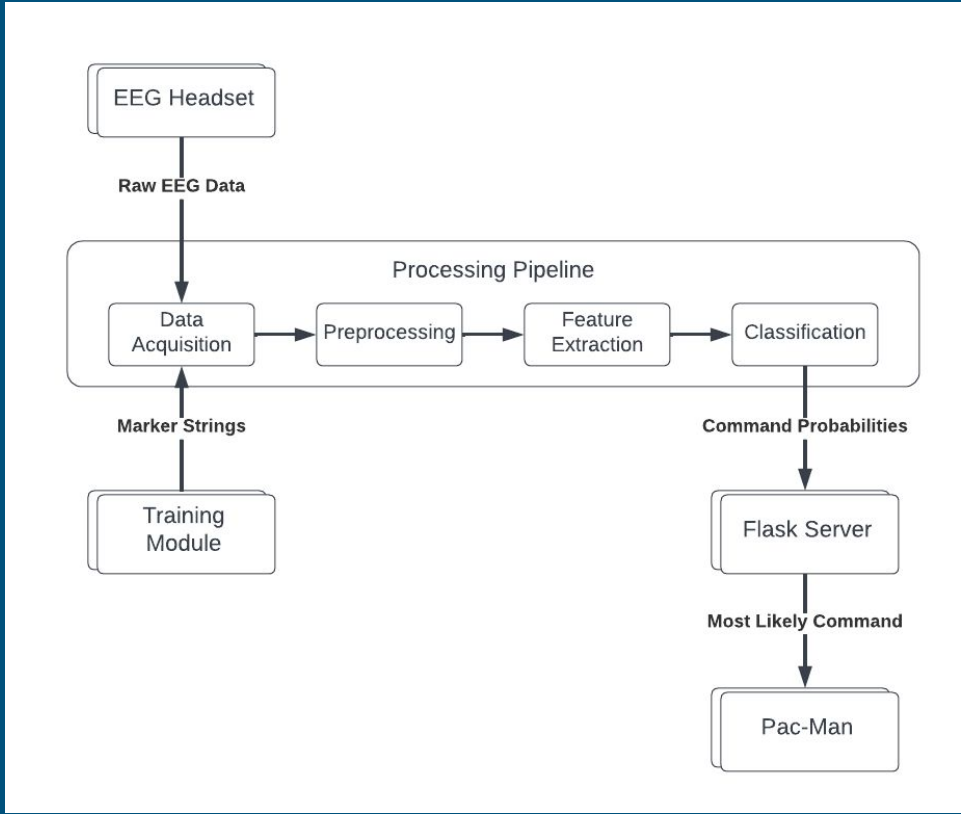


Fig. 6 - A diagram of the MindGames system architecture

Collecting Data

- EEG Signals
  - Spike electrodes
  - OpenBCI Daisy Biosensing Board
  - OpenBCI Ultracortex Mk 4 headset
- ML Training Labels
  - Training module
- Data synchronized, sent to processing pipeline

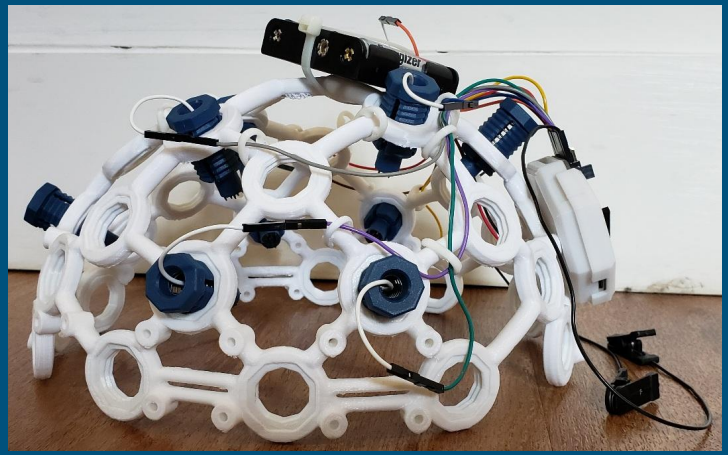


Fig. 7 - Ultracortex headset, profile view

EEG Processing Pipeline

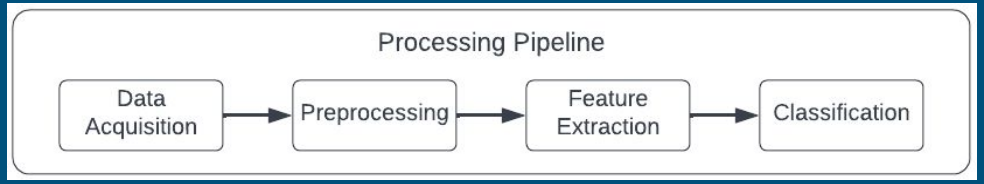


Fig. 8 - MindGames EEG processing pipeline simplified view

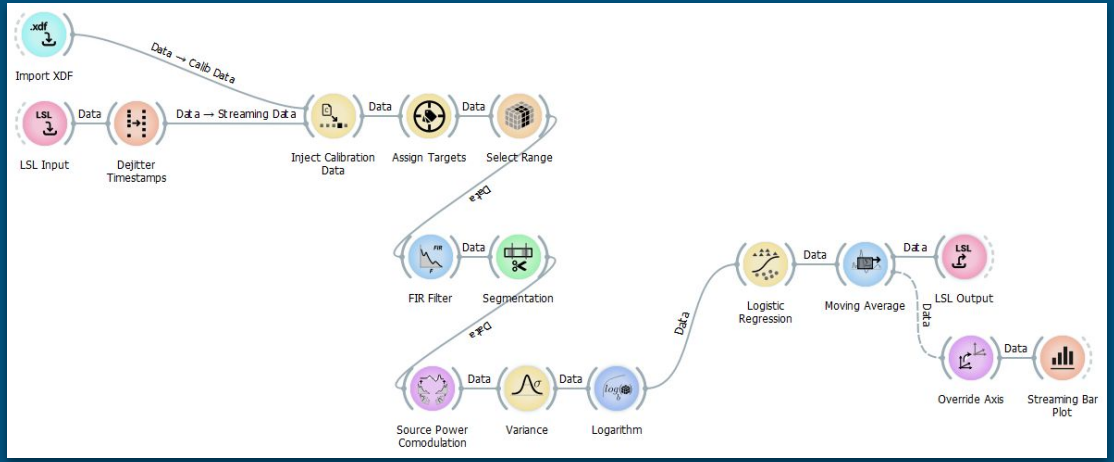


Fig. 9 - MindGames EEG processing pipeline detailed view

EEG Processing Pipeline

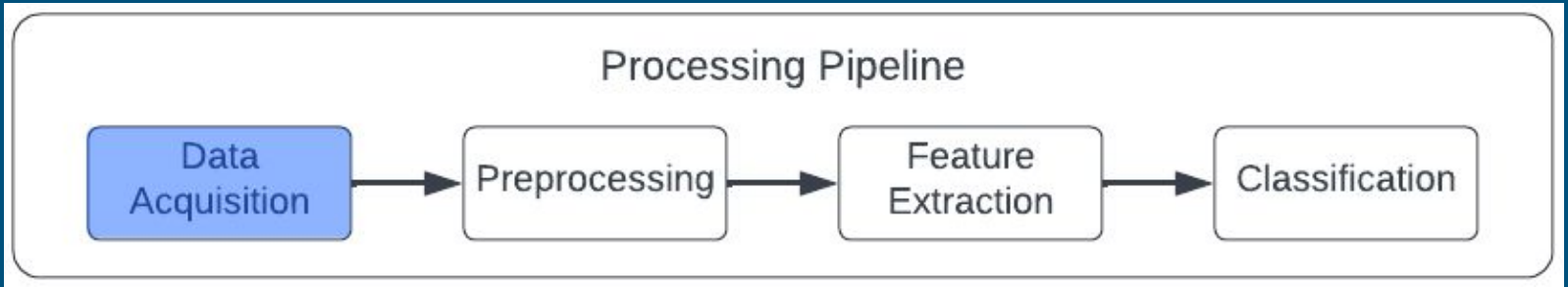
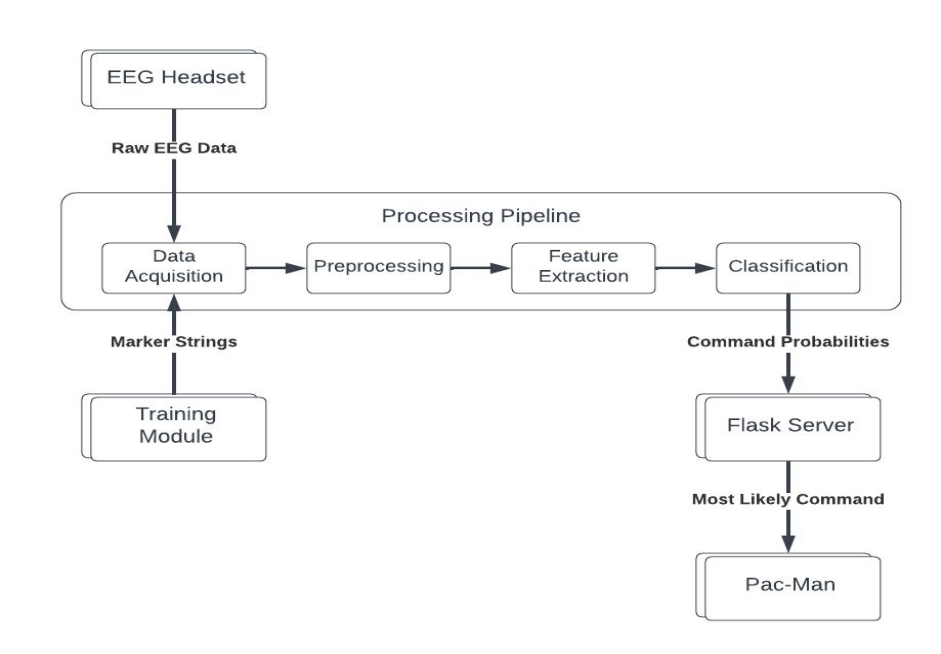


Fig. 10 - EEG pipeline data acquisition component

Using Commands



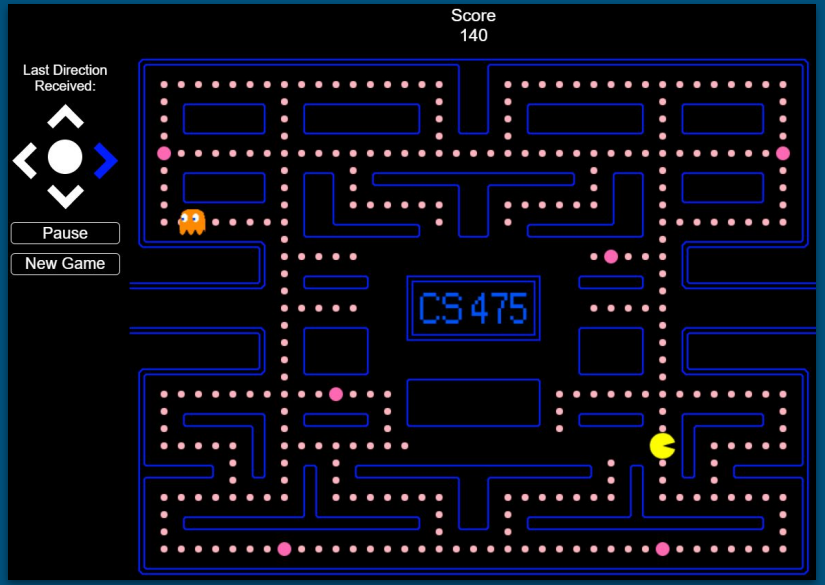


Fig. 14 - Pac Man in action

Accomplished Work

- Pac-Man web-frontend
  - Javascript, HTML, CSS
- Flask backend
  - Python
- OpenBCI Specific Development and Communication Tools

Challenges

- Training time
- Hardware Limitations
- Noise and unrelated spikes
- Thought to Action Latency

Fig. 15 - OpenBCI Headset with Additional Nodes

Lessons Learned and Future Work

- Upgrade our EEG hardware
- Fine-tune our noise reduction and ML classification techniques
- Transition from classifying movements to classifying imagined movements (motor imagery)

Questions?