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

1. Brain activity (signals) sent to computer
2. Computer processes signals to commands
3. Commands sent to software and executed

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

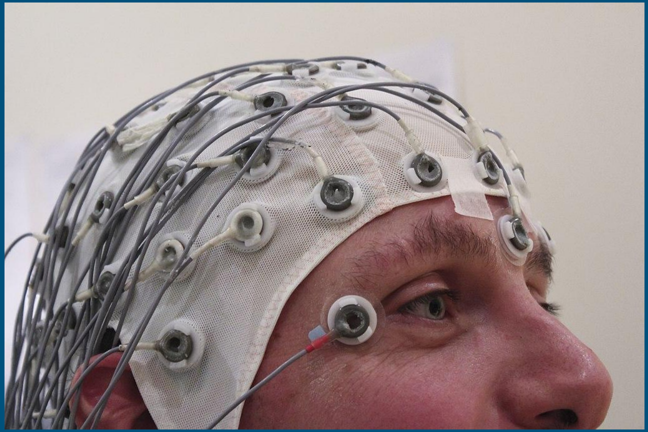

Fig. 2 - An example of EEG hardware

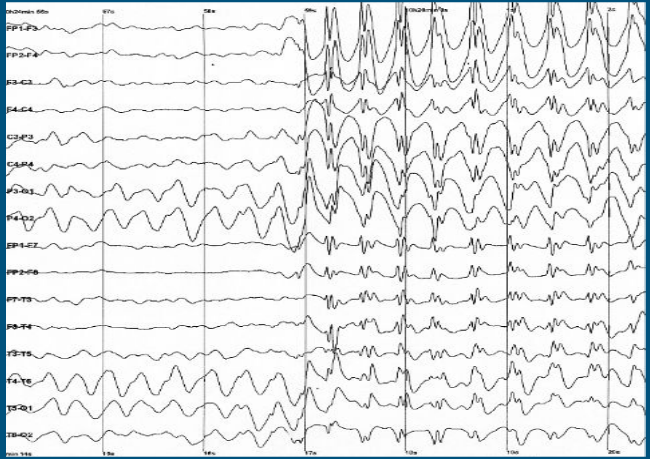

Fig. 3 - Sample EEG output

Building a Neural Fingerprint

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

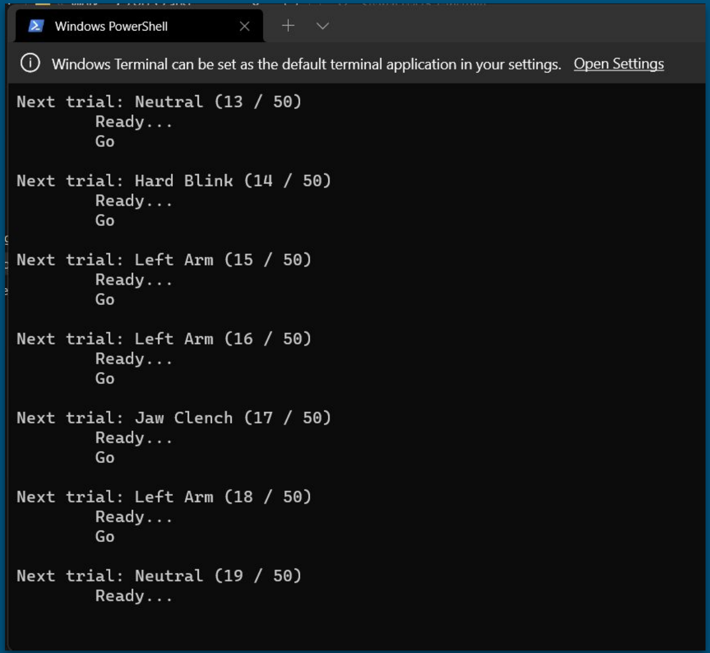

Fig. 4 - Calibration module collecting training data

Command Classification

- Enough examples for ML model to classify patterns on its own

- Probabilistic:
  - Determines **most likely command** given pattern

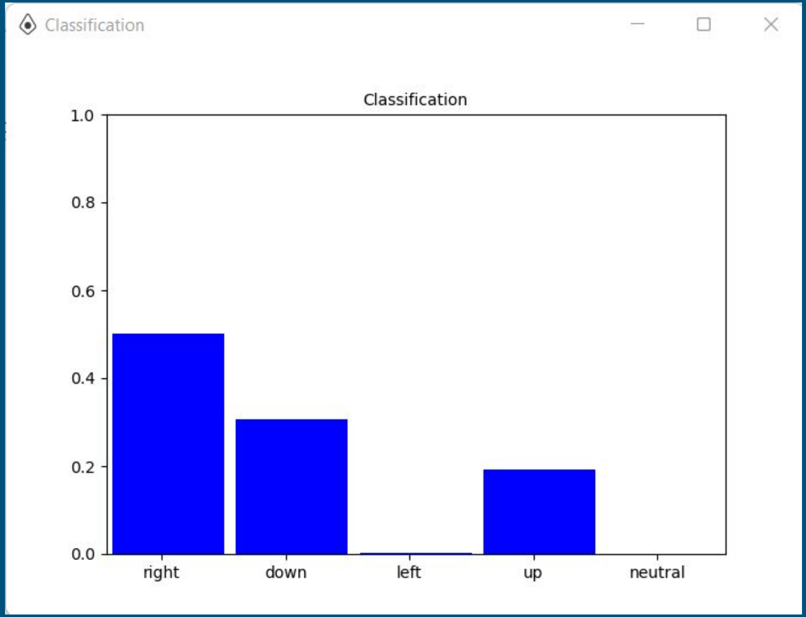

Fig. 5 - Visualization of model's classification output

System Overview

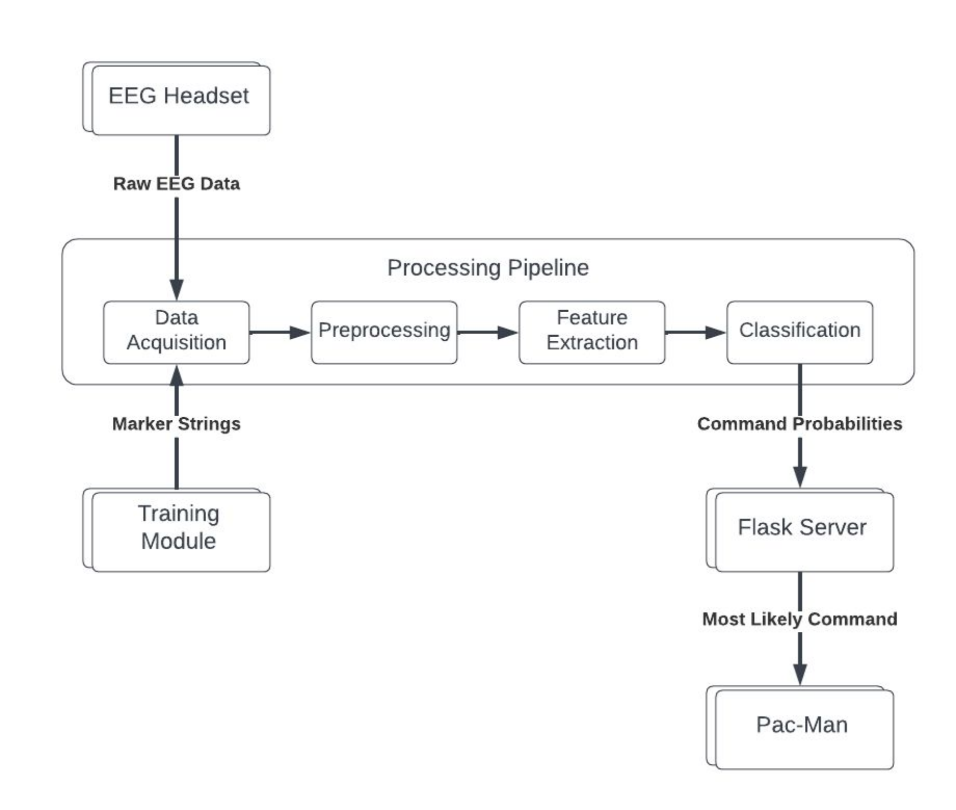

Fig. 6 - A diagram of the MindGames system architecture

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

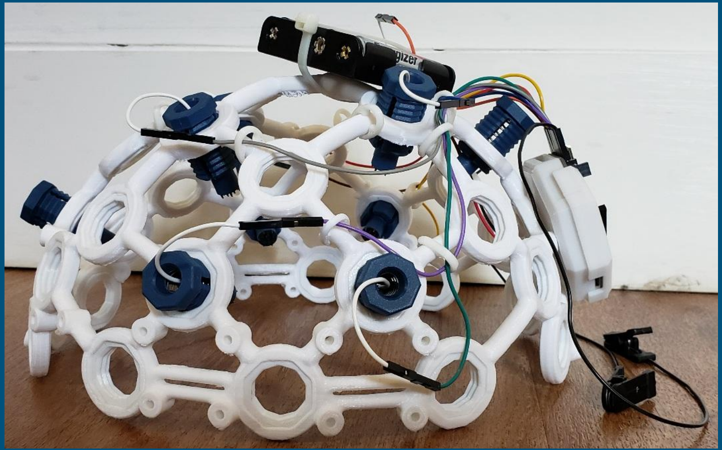

Fig. 7 - Ultracortex headset, profile view

EEG Processing Pipeline

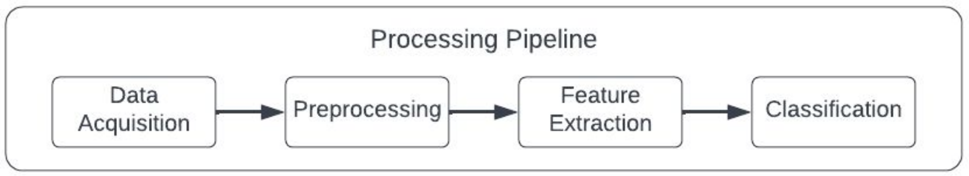

Fig. 8 - MindGames EEG processing pipeline simplified view

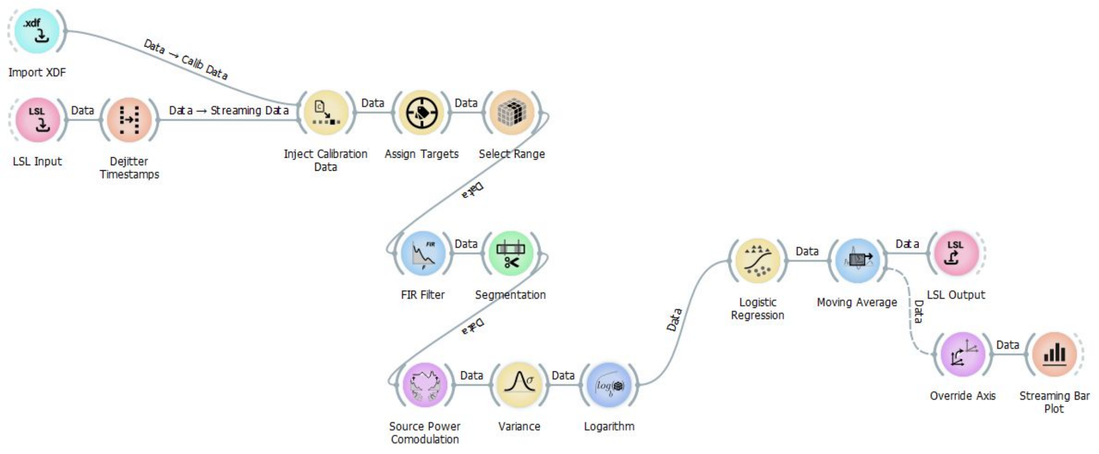

Fig. 9 - MindGames EEG processing pipeline detailed view

EEG Processing Pipeline - Data Acquisition

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

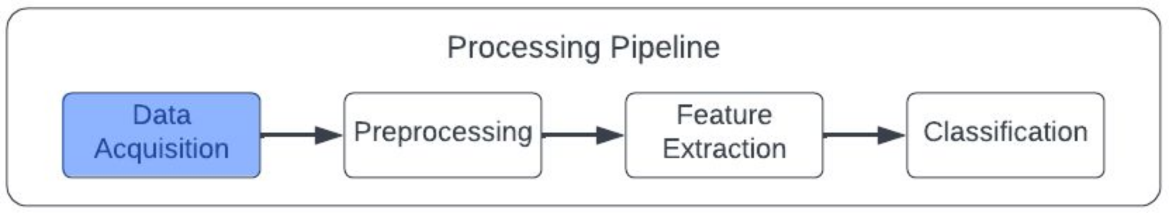

Fig. 10 - EEG pipeline data acquisition component

EEG Processing Pipeline - Preprocessing

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

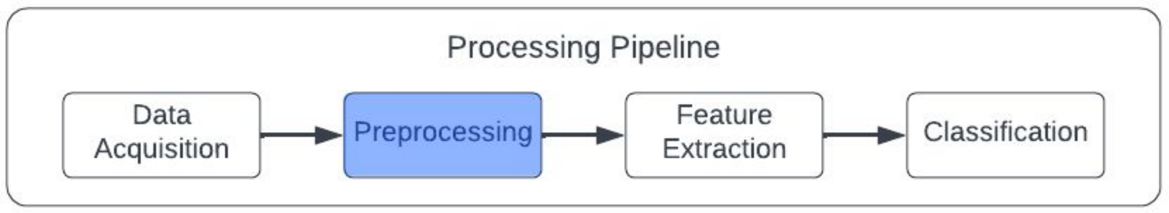

Fig. 11 - EEG pipeline preprocessing component

EEG 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

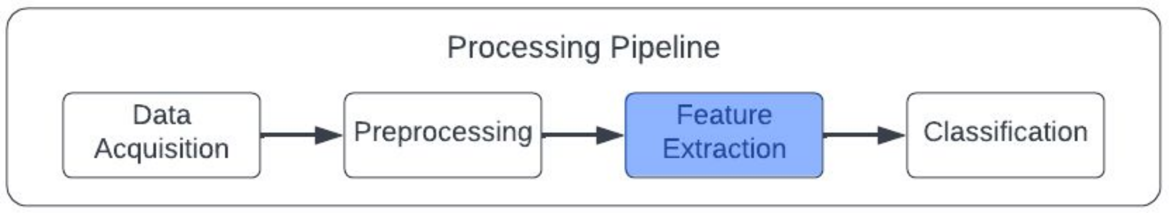

Fig. 12 - EEG pipeline feature extraction component

EEG Processing Pipeline - Classification

- Use extracted features to determine command probabilities
- Maintain a moving average of probabilities to reduce noise
- Output list of probabilities

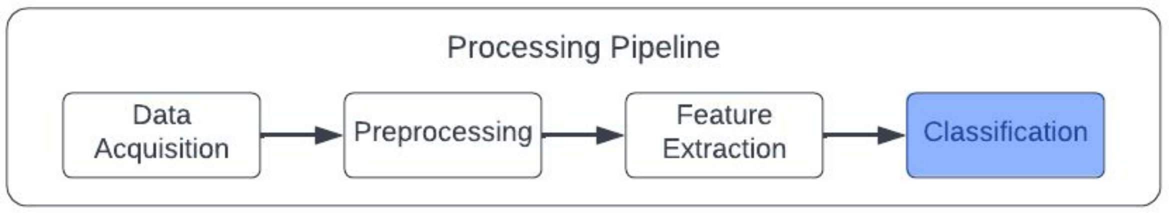

Fig. 13 - EEG pipeline classification component

Using Commands

- List of probabilities received by Flask backend
- Highest probability extracted
- Corresponding command sent to Pac-Man webfront
- Command executed in-game

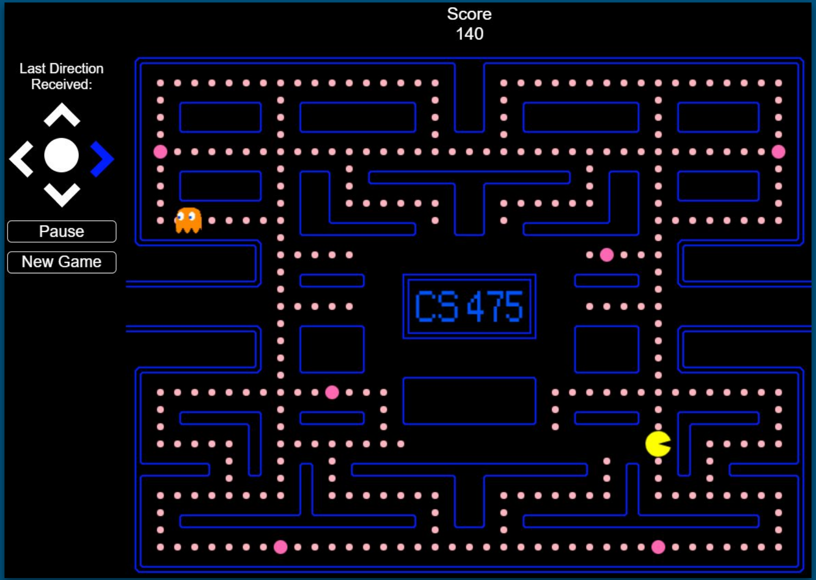

Fig. 14 - Pac Man in action

Accomplished Work

- Pac-Man webfront
  - Javascript, HTML, CSS
- Flask backend
  - Python
- Node-based EEG processing pipeline
  - NeuroPype Pipeline Designer
- Communication between components
  - Lab Streaming Layer, WebSocket API

Challenges - Hardware

- OpenBCI Ultracortex headset
  - 8 electrodes (professional: 128)
  - Difficult to secure
  - Small movements cause it to shift position
  - Electrodes move with headset

- RESULT:
  - Low quality data (and not enough of it)

Challenges - Software

- Training time: 8 minutes
  - Results subject to hardware challenges
- Noise and unrelated spikes (e.g., blinking, swallowing)
  - Must be filtered out without affecting relevant signals
- Thought to Action
  - Minimal latency for the game to feel responsive

Lessons Learned

- ML requires a lot of data
- EEG signals are noisy
- EEG hardware limitations
- Powerful computer needed for real-time signal processing

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?