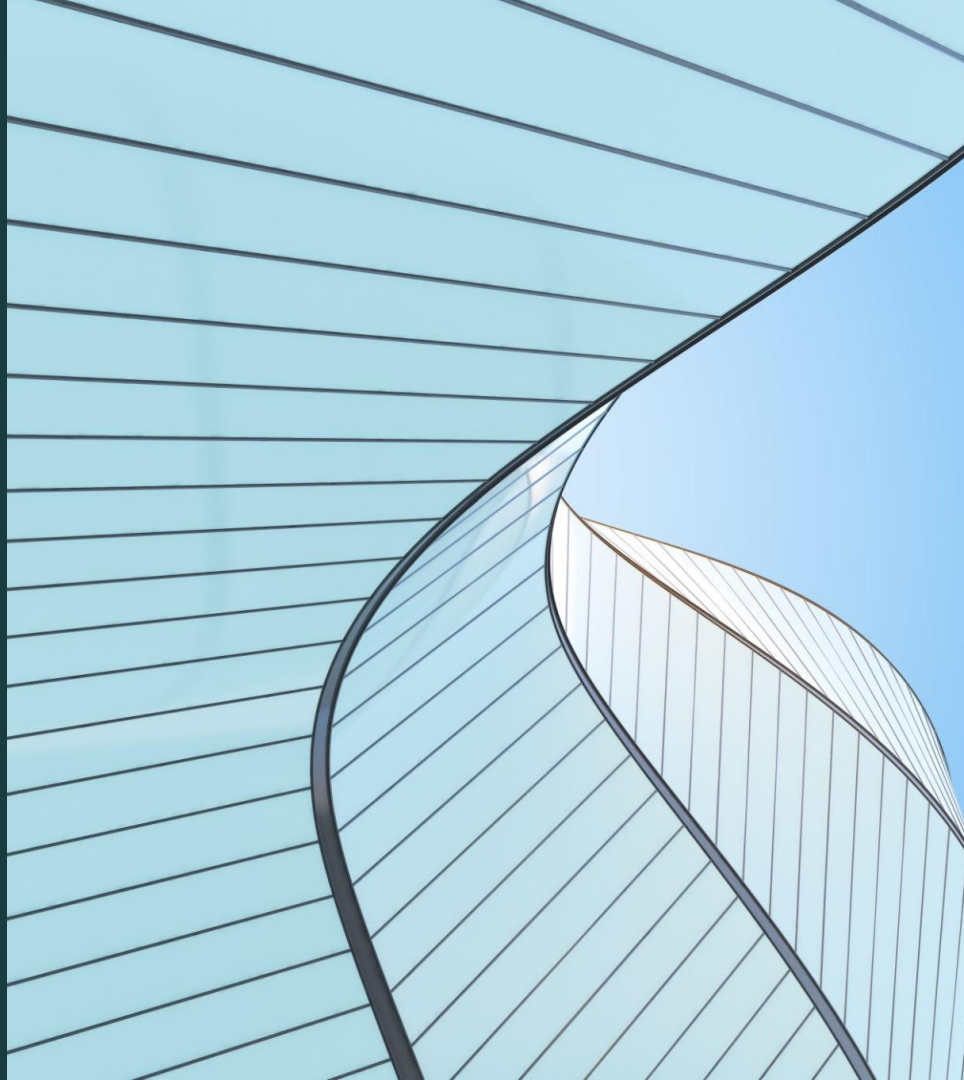
Team 5
24 October 2025

SmartPEN

EMG powered Art Simulator and image analysis.



Content *overview*

1 Motivation

2 Theoretical Design

3 Design

4 Results

5 Future Work

6 Q&A

Team Members

Anisa Callis

MS HCI at RIT

Connor Liu

PhD in Sustainability at RIT

Stephanie Patterson

MS HCI at RIT

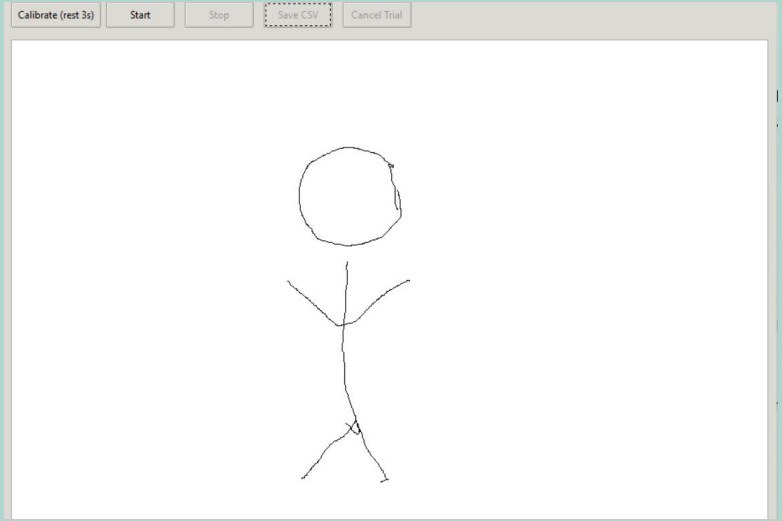
Chirayu Salgarkar

PhD in Mechanical and Industrial Engineering at RIT

Goal: Implement gesture recognition for development of drawing tool.

Secondary Goal: Image recognition of drawn material for categorization and gestures for image details.

THE PROBLEM



Drawing created from final deliverable.

Research Question:

Can the Myo armband function as a preliminary drawing tool?

Theoretical Protocol

The Myo Armband can classify gestures for classification, as well as use IMU data to draw!

01

Train classifier model for gestures

Use *myo-emg-dataset* from Kaggle

02

Develop drawing tool for sketching

Shape_capture method, similar to MS Paint

03

Incorporate gestures + drawing tool to make sketches with various thicknesses

External: save basic sketches to png images

04: Future

Use protocol to convert front, side and top view to isometric drawing

Benefit: construct images for preliminary product conception

myo-eng-dataset:

- 5000 samples, 30 readings/sample
- Format: numpy ndarray [N, 30, 8] (#number of samples, readings per sample, channels)
- SVM model to train data, test sample size = 0.2, train sample size = 0.8

Result: overall accuracy of trained set is 0.92 (from prior model)

4 gestures used:

A. 0-Neutral Gesture
B. 1-Flexion
C. 2-Extension
D. 7-Fist

Column 1	Precision	Recall	f1-score	Gesture Count
0	0.89	0.96	0.92	558
1	0.94	0.86	0.9	187
2	0.97	0.86	0.91	187
7	0.96	0.91	0.93	187

Developed SVM model and used the result to predict the class of the most recent 30 samples in real time.

Real-time data needed to be abs() and multiplied by 10 to match training data.

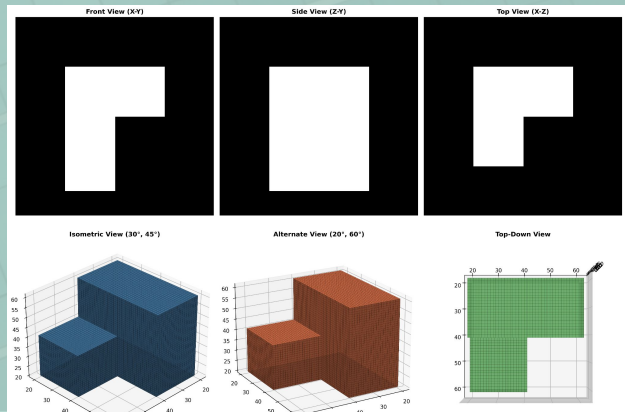
Results: Videos

Future application:

Three view sketch (gesture used to change view), automatically converted to isometric view

Tools:

1. Numpy, store 3D coordinates
2. MatPlotLib: 3D plotting (mpl_toolkits.mplot3d), Poly3DCollection for 3D faces



Thank you