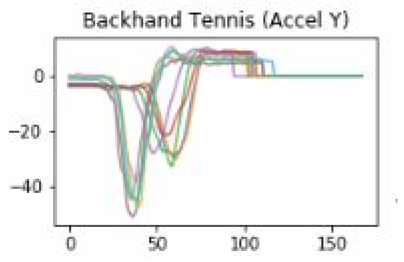
D. Tim Sawyer
A2: Gesture Recognizer

Code
https://github.com/tisawyer/ubicomp/blob/master/GestureRecognizer/Sawyer%20A2%20Gesture%20Recognizer%20Final.ipynb

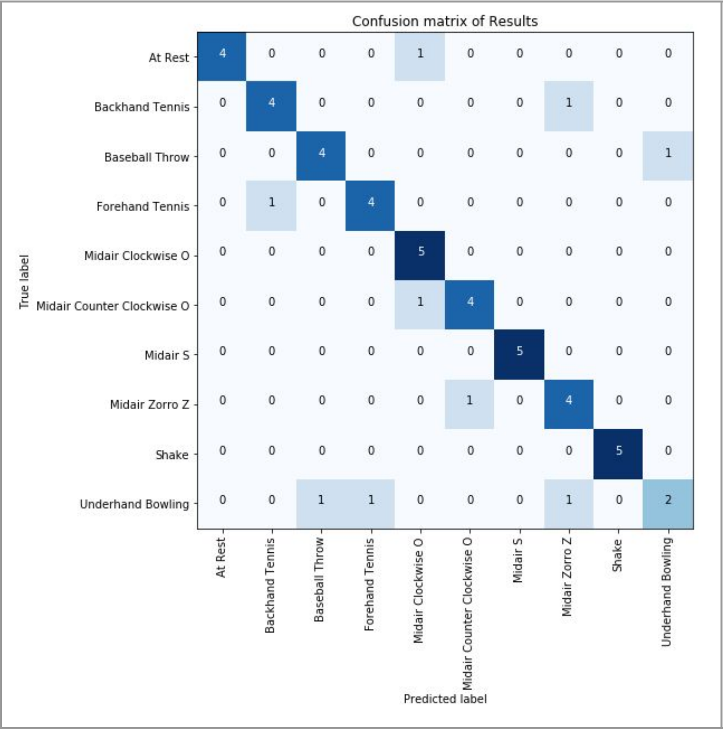
Data collection
I attempted to create a more difficult dataset by gathering gestures sets from 2 users. Each gesture was performed and recorded ten times. Five of these were performed by me, and five were performed by my wife. We tried to perform the gestures as identically as possible, but in the data visualizations that I created, you can clearly see the two different clusters. For example:

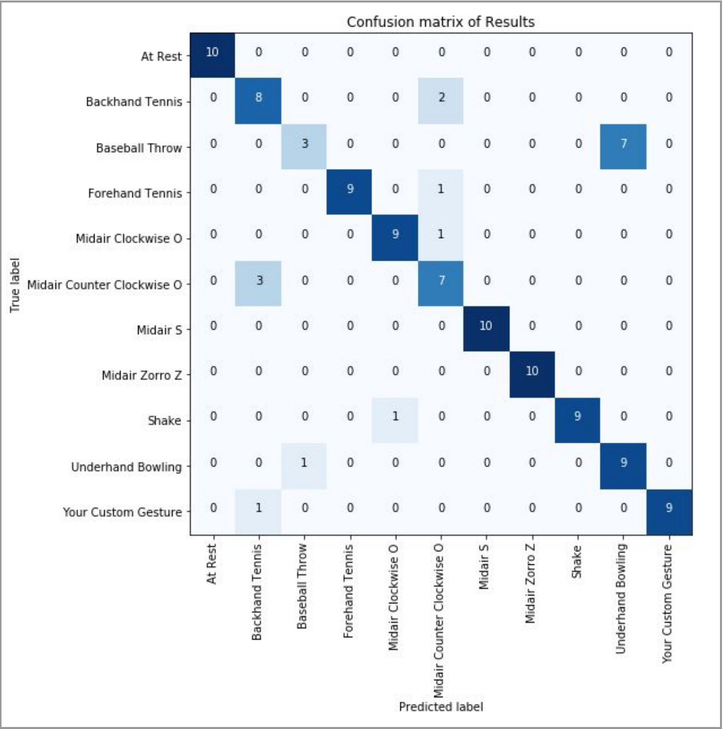


For the the custom gesture, we performed a “sword thrust” movement.

Shape Matching
For the shape matching I simply took the Euclidean distance between the incoming gesture and every gesture in the training folds. The gesture in the training folds with the lowest Euclidean distance was my prediction of the incoming gesture. My results for each dataset were as follows:

Results with Jon's dataset At Rest: 4/5 (80.0%) Backhand Tennis: 4/5 (80.0%) Baseball Throw: 4/5 (80.0%) Forehand Tennis: 4/5 (80.0%) Midair Clockwise O: 5/5 (100.0%) Midair Counter Clockwise O: 4/5 (80.0%) Midair S: 5/5 (100.0%) Midair Zorro Z: 4/5 (80.0%) Shake: 5/5 (100.0%) Underhand Bowling: 2/5 (40.0%) Total Euclidean distance accuracy 82.0%	Results with My dataset At Rest: 10/10 (100.0%) Backhand Tennis: 8/10 (80.0%) Baseball Throw: 3/10 (30.0%) Forehand Tennis: 9/10 (90.0%) Midair Clockwise O: 9/10 (90.0%) Midair Counter Clockwise O: 7/10 (70.0%) Midair S: 10/10 (100.0%) Midair Zorro Z: 10/10 (100.0%) Shake: 9/10 (90.0%) Underhand Bowling: 9/10 (90.0%) Your Custom Gesture: 9/10 (90.0%) Total Euclidean distance accuracy 84.54545454545455%

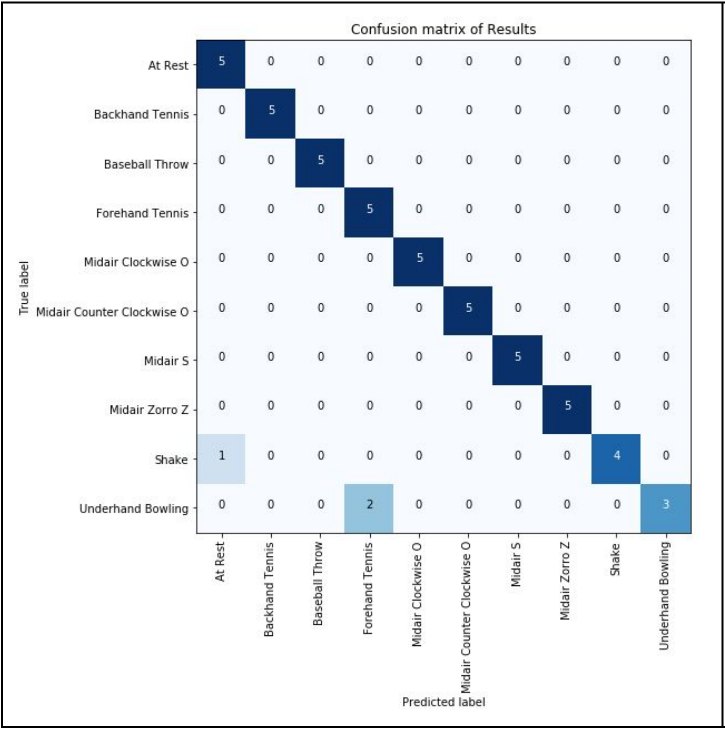


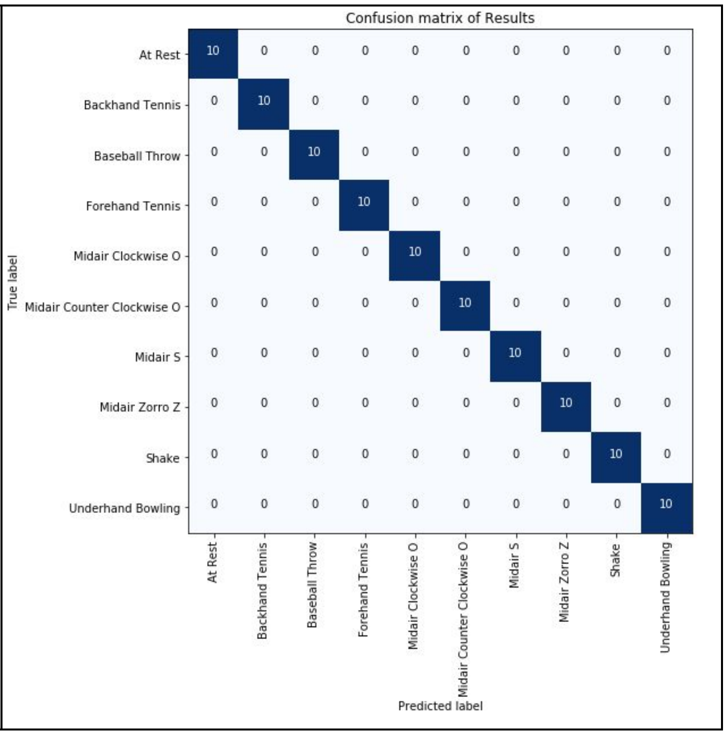


Model-based classification approach
My model based classification approach was to use an SVM. The features that I extracted were based roughly off of this article:
https://link-springer-com.offcampus.lib.washington.edu/content/pdf/10.1007%2F978-3-642-02830-4.pdf

Entropy turned out to a very feature good feature to extract. Entropy of the signal was calculated after performing a Fourier transform on the signal. I attempted to use the energy of the signal also but this significantly decreased the accuracy. For each features, I ended up not using magnitude, but rather extracted features from each axis (x,y,z) separately. Also, I attempted to use a butterworth low pass filter over each signal, but using this did not noticeably improve my accuracy. With Jon's dataset, I was able to achieve as high 94% accuracy on some runs. With my dataset, I was able to achieve 100% accuracy!

SVM Results with Jon's dataset	SVM Results with My dataset
At Rest: 5/5 (100.0%)	At Rest: 10/10 (100.0%)
Backhand Tennis: 5/5 (100.0%)	Backhand Tennis: 10/10 (100.0%)
Baseball Throw: 5/5 (100.0%)	Baseball Throw: 10/10 (100.0%)
Forehand Tennis: 5/5 (100.0%)	Forehand Tennis: 10/10 (100.0%)
Midair Clockwise O: 5/5 (100.0%)	Midair Clockwise O: 10/10 (100.0%)
Midair Counter Clockwise O: 5/5 (100.0%)	Midair Counter Clockwise O: 10/10 (100.0%)
Midair S: 5/5 (100.0%)	Midair S: 10/10 (100.0%)
Midair Zorro Z: 5/5 (100.0%)	Midair Zorro Z: 10/10 (100.0%)
Shake: 4/5 (80.0%)	Shake: 10/10 (100.0%)
Underhand Bowling: 3/5 (60.0%)	Underhand Bowling: 10/10 (100.0%)
	Your Custom Gesture: 10/10 (100.0%)
Total SVM classifier accuracy 94.0%	
	Total SVM classifier accuracy 100.0%





Learnings

I was quite surprised at first that both the Euclidean distance and SVM approaches were more accurate with my dataset. However, I think this does show how more data is generally beneficial for building an accurate model. It was impressive to see how many features the SVM approach could handle. I ended up only using 21 features, but some of the papers mentioned using several hundred. I also really enjoyed using Jupyter notebook, and was delighted to see how easy it was to train and use a machine learning model in Python.