Journal Report 14
1/6/20 - 1/10/20
Sophia Wang
Computer Systems Research Lab
Period 1, White

Daily Log

Monday January 6

I began finding videos online for the different moves we're using. Additionally, I began cutting them so it's just the desired move.

Tuesday January 7

I recorded my own versions of the moves to expand our pool of test videos. I also looked into implementing dynamic time warping.

Thursday January 10

I helped write the kmeans code to classify poses and cut the videos I recorded into shorter videos for video processing.

Timeline

Date	Goal	Met
12/16	Classify pose using k-nn	No, but we have all the angle classifying data, we will need to gather more data as well
1/6	Start on Dynamic Time Warping code and work on classifying a move	No
1/13	Gather test data to train the kmeans	Yes, we have both online and videos filmed to train
1/20	Finish and train kmeans	
1/27	Begin working on Dynamic Time Warping code and classifying a move	

Reflection

Last week, I was mostly focused on gathering data to use moving forward. My partner was working on the kmeans code, which we are writing on colab. Our switch back to colab is due to it's increased processing power, however for accessibility, we will probably make a python version for our github.

The kmeans will group the sets of body angles from the videos and decide on poses. Below shows our kmeans method to find the optimal number of clusters. This is currently with randomized data, but this week, with the new test data, we will be applying this to poses.

```
wcss = []
for i in range(1, 11): #trying out different numbers of clusters
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_) #inertia_ tells you how good it is
plt.plot(range(1, 11), wcss) #this is all just plotting the results stuff
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```