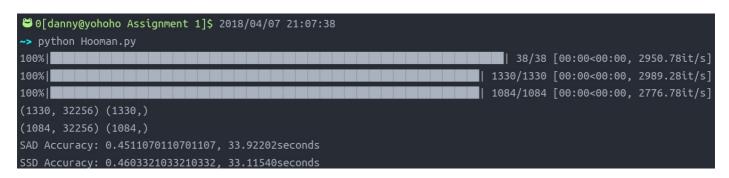# Nearest Neighbor Search

Method Description

1. Load data path
2. Load data and labels
3. Find nearest neighbor with sum of absolute distance between test data and training data
   - Implemtation: scipy.spatial.distance.cdist

4. Find nearest neighbor with sum of square distance between test data and training data
   - Implemtation: scipy.spatial.distance.cdist

Results

```
0[danny@yohoho Assignment 1]$ 2018/04/07 21:07:38
~> python Hooman.py
100%|                                                    | 38/38 [00:00<00:00, 2950.78it/s]
100%|                                                    | 1330/1330 [00:00<00:00, 2989.28it/s]
100%|                                                    | 1084/1084 [00:00<00:00, 2776.78it/s]
(1330, 32256) (1330,)
(1084, 32256) (1084,)
SAD Accuracy: 0.4511070110701107, 33.92202seconds
SSD Accuracy: 0.4603321033210332, 33.11540seconds
```

SAD: 0.4511 in 33.92 seconds
SSD: 0.4603 in 33.12 seconds

Discussion

During this assignment, I've tried 3 kinds of ways computing the results. The 3 ways are for loop, numpy broadcasting, and cdist from scipy in the order from slow to fast. I also learned something really important: The generator mechanism in Python is something really passive. Hence, generators should be used wisely, as they would slow down the whole process as a side effect.