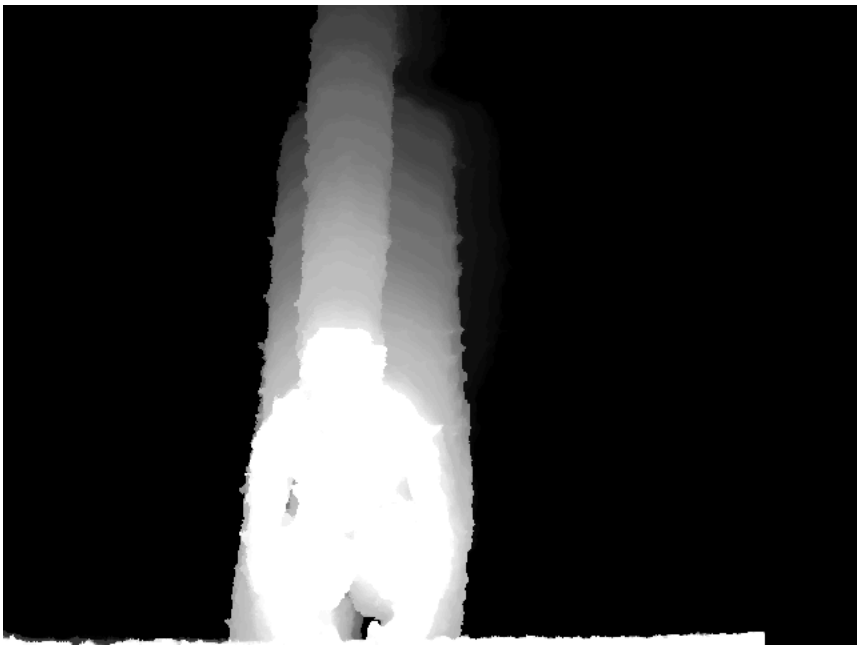
PS5
Shaurye Aggarwal

1.1)

Both Arms

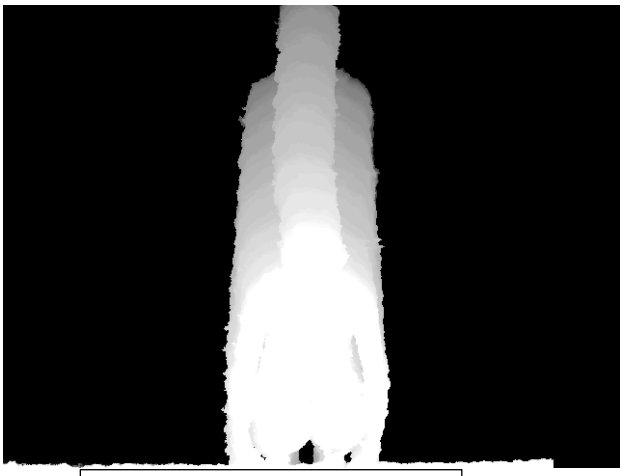
Right Kick



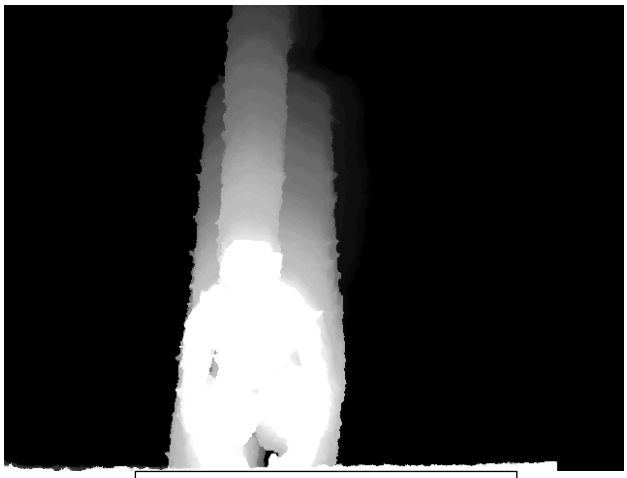
Crouch

1.4)

Target Activity: Crouch | All MHI Index = 5

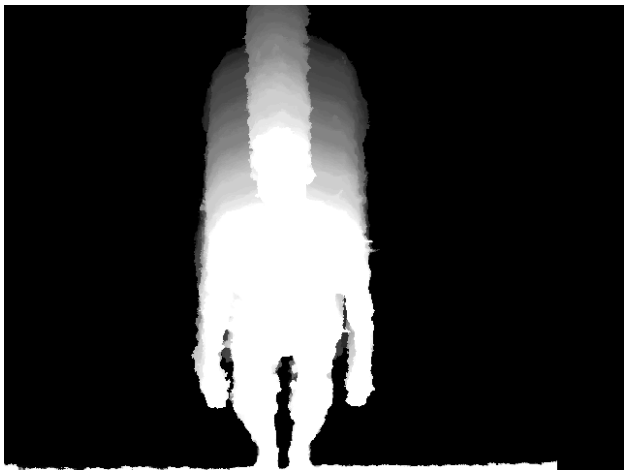


1st Nearest Neighbor



2nd Nearest Neighbor

3rd Nearest Neighbor



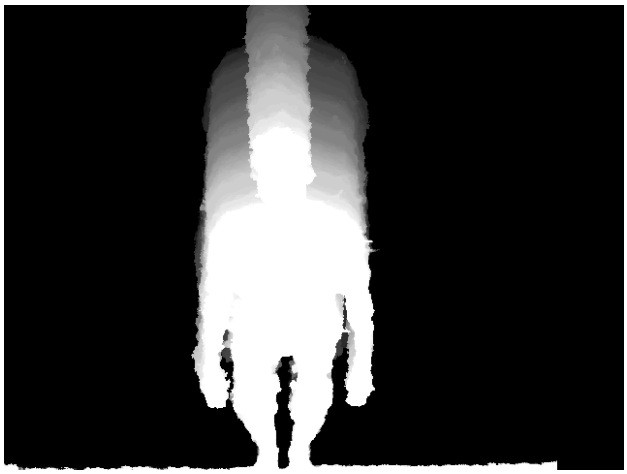
4th Nearest Neighbor

Target Activity: Right Kick | All MHI Index = 16

1st Nearest Neighbor

2nd Nearest Neighbor

3rd Nearest Neighbor



4th Nearest Neighbor

5)

	botharms	crouch	leftarmup	punch	rightkick
botharms	4	0	0	0	0
crouch	0	4	0	0	0
leftarmup	0	0	4	0	0
punch	0	0	1	3	0
rightkick	0	0	0	1	3

Mean Recognition Rates Per Class

botharms : 1.0
crouch : 1.0
leftarmup : 1.0
punch : 0.75
rightkick : 0.75

As it can be seen “botharms”, “crouch”, “leftarmup" achieve near perfect accuracy having 1.0 mean recognition rate. I feel this is because if you look at the training Motion History Images for each one of them they all look really similar to each other and hence the lowest distance from the hu moments of the test instance is actually to the hu moments of the same class.

The most confused classes – “rightkick” and “punch” – miss out on one case each. “punch-p2-1.png” is recognized as “leftarmup” as it’s rotation invariant hu moments have the least Normalized Euclidean Distance to “leftarm-up-p2-2.png”. I think the two images are pretty similar so as to be confused because the left arm up creates a radial region of gray on the left side of the body. If you notice the MHI for the punch, it has a radial gray region on the right. The body and head in both are tilted at an angle and the other arm is rather stationary for both the MHIs. The right kick instance is surprisingly matched with a punch instance. If looked at the distance, the actual correct labels are in fact surprisingly at greater distances than many other classes.

Extra Credit

2) For the extra credit, I tried 3 various stratergies:

1) Manipulate background subtraction threshold

To reduce the threshold by 300 and increase it to 40000 and hence now the pixels having a value greater than 39500 are considered as background. This I thought might help reduce the noise I have in the current image or add more detail to MHIs respectively. However in both the cases, the results were the same as above. The MHIs and hu Vectors for the 39500 threhold can be accessed using “allMHIs40000.npy” and “huVectors40000.npy”.

	botharms	crouch	leftarmup	punch	rightkick
botharms	4	0	0	0	0
crouch	0	4	0	0	0
leftarmup	0	0	4	0	0
punch	0	0	1	3	0
rightkick	0	0	0	1	3

Mean Recognition Rates Per Class

botharms : 1.0
crouch : 1.0
leftarmup : 1.0
punch : 0.75
rightkick : 0.75

2) Mode

Instead of using 1–NN I tried using mode of the first k = 4 labels because I thought if there are other close matches they should be considered too. More matches from the same label would mean that the test instance has higher chance of belonging to that particular label. Here the accuracy went further down for botharms, leftarmup, but remained same for rightkick, crouch and punch.This maybe because sometimes more number of other instances have hu moments closer and the k for calculating the mode would change from case to case. This function is in predictAction.py as predictActionMode function.

	botharms	crouch	leftarmup	punch	rightkick
botharms	2	2	0	0	0
crouch	0	3	1	0	0
leftarmup	0	0	1	3	0
punch	0	0	1	3	0
rightkick	0	0	1	0	3

```
botharms  :  0.75
crouch  :  1.0
leftarmup  :  0.25
    punch  :  0.75
 rightkick  :  0.75
```

3) Mean

Instead of using 1-NN I tried using mean of the first k = 4 labels to see what change it produces. Here the accuracy went further down for botharms, leftarmup, and right kick but it went up for punch. This function is in predictAction.py as predictActionMean function.

	botharms	crouch	leftarmup	punch	rightkick
botharms	2	2	0	0	0
crouch	0	4	0	0	0
leftarmup	0	0	0	4	0
punch	0	0	0	4	0
rightkick	0	0	1	3	0

Mean Recognition Rates Per Class

```
botharms  :  0.5
crouch  :  1.0
leftarmup  :  0.0
punch  :  1.0
rightkick  :  0.0
```