***Speed and Acceleration***

**Speed**

For getting speed, we require the location and frame index of the last two frames in which the object appears.

self.get\_speed(

self.track\_id\_centroid[i[4]]["centroids"][-1],

self.track\_id\_centroid[i[4]]["centroids"][-2],

(

self.track\_id\_centroid[i[4]]["frame\_indexes"][-1]

- self.track\_id\_centroid[i[4]]["frame\_indexes"][-2]

)/ 30 )

Here -1 represents the last centroid of that object and -2 represents the second last centroid similarly for frame index.

def get\_speed(

self,

centroid1,

centroid2,

time,

):

centroid1\_geo = localize(centroid1)

centroid2\_geo = localize(centroid2)

dist = self.distance(centroid1\_geo, centroid2\_geo)

return dist / time

For distance, we localize the centroid from pixel coordinates to lat/long coordinate

We can get time, by taking the difference of frame index and divide it by the fps of the video.

def distance(self, c1, c2):

lat1, lon1 = c1

lat2, lon2 = c2

p = pi / 180

a = (0.5 - cos((lat2 - lat1) \* p) / 2

+ cos(lat1 \* p) \* cos(lat2 \* p) \* (1 - cos((lon2 - lon1) \* p))

/2)

return 12742 \* asin(sqrt(a)) \* 1000

After the object disappears we take the average of all the speeds.

**Acceleration**

For getting acceleration, we require the speed and frame index of the last two frames in which the object appears.

If the length of the speed list is greater than 2 then we are able to calculate acceleration else we append 0 in the acceleration list.

self.get\_acceleration(

self.track\_id\_centroid[i[4]]["speeds"][-2],

self.track\_id\_centroid[i[4]]["speeds"][-1],

(

self.track\_id\_centroid[i[4]]["frame\_indexes"][-1]

-self.track\_id\_centroid[i[4]]["frame\_indexes"][-2]

) /30 ,)

def get\_acceleration(self, speed1, speed2, time):

return (speed2 - speed1) / time

After the object disappears, we take the average of all the accelerations.