indian dataset

July 29, 2018

0.0.1 Importing the libraries

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib
        import os
        import subprocess
```

0.0.2 The base path of the folder where the data is stored.

```
In [12]: path='C:\\Users\\asd\\Desktop\\sort-tracker\\India 2\\500 frames'
```

0.0.3 Example DataFrame

```
In [13]: readFile=pd.read_csv(path+"/1/2/002.csv")
         print(readFile)
```

```
      frameNumber        label  id  x_TL  y_TL  x_BR  y_BR  isOccluded  \
0               0          Car   0   332   255   523   160           0
1               0          Car   1    40   186    75   141           0
2               0  Two-Wheeler   2   612   233   663   198           0
3               0   Pedestrian   3   810   207   823   178           0
4               0   Pedestrian   4   800   207   811   175           0
5               1          Car   0   315   255   510   158           0
6               1          Car   1    31   186    66   141           0
7               1  Two-Wheeler   2   603   233   656   196           0
8               1   Pedestrian   3   804   207   817   177           0
9               1   Pedestrian   4   795   207   806   174           0
10              2          Car   0   298   256   498   156           0
11              2          Car   1    23   186    58   141           0
12              2  Two-Wheeler   2   595   233   650   195           0
13              2   Pedestrian   3   798   207   812   176           0
14              2   Pedestrian   4   789   207   801   173           0
15              3          Car   0   281   257   486   155           0
16              3          Car   1    15   186    50   141           0
17              3  Two-Wheeler   2   586   233   643   193           0
18              3   Pedestrian   3   793   207   807   176           0
19              3   Pedestrian   4   784   207   796   173           0
```

20	4	Car	0	264	258	474	153	0
21	4	Car	1	7	186	42	141	0
22	4	Two-Wheeler	2	578	234	637	192	0
23	4	Pedestrian	3	787	207	802	175	0
24	4	Pedestrian	4	778	207	790	172	0
25	5	Car	0	247	259	462	151	0
26	5	Two-Wheeler	2	569	234	630	190	0
27	5	Pedestrian	3	782	207	796	174	0
28	5	Pedestrian	4	773	207	785	172	0
29	6	Car	0	230	260	450	150	0
...	...	...	..	...	...	...	...	...
1419	492	Car	15	592	192	631	160	0
1420	492	Car	16	662	189	682	158	0
1421	493	Car	9	682	207	733	161	0
1422	493	Car	14	519	215	592	161	0
1423	493	Car	15	591	192	631	160	0
1424	493	Car	16	662	189	682	158	0
1425	494	Car	9	682	207	733	161	0
1426	494	Car	14	517	216	591	161	0
1427	494	Car	15	591	193	630	160	0
1428	494	Car	16	662	189	682	158	0
1429	495	Car	9	682	207	733	161	0
1430	495	Car	14	515	216	589	161	0
1431	495	Car	15	590	193	630	160	0
1432	495	Car	16	662	189	682	158	0
1433	496	Car	9	683	207	734	161	0
1434	496	Car	14	512	217	588	161	0
1435	496	Car	15	589	193	630	160	0
1436	496	Car	16	662	189	682	158	0
1437	497	Car	9	683	209	734	163	0
1438	497	Car	14	510	217	586	161	0
1439	497	Car	15	589	193	629	160	0
1440	497	Car	16	662	189	682	158	0
1441	498	Car	9	683	211	734	165	0
1442	498	Car	14	508	218	585	161	0
1443	498	Car	15	588	193	629	160	0
1444	498	Car	16	662	189	682	158	0
1445	499	Car	9	684	213	735	167	0
1446	499	Car	14	506	219	584	161	0
1447	499	Car	15	588	194	629	161	0
1448	499	Car	16	662	190	683	158	0

	toInterpolate	isInterpolated
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0

5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1
13	1	1
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	1	1
20	1	1
21	0	0
22	1	1
23	1	1
24	1	1
25	1	1
26	1	1
27	1	1
28	1	1
29	1	1
...	...	...
1419	1	1
1420	1	1
1421	1	1
1422	1	1
1423	1	1
1424	1	1
1425	1	1
1426	1	1
1427	1	1
1428	1	1
1429	1	1
1430	1	1
1431	1	1
1432	1	1
1433	1	0
1434	1	1
1435	1	1
1436	1	1
1437	1	1
1438	1	1
1439	1	1
1440	1	1

```
1441              1               1
1442              1               1
1443              1               1
1444              1               1
1445              1               0
1446              1               0
1447              1               0
1448              1               0

[1449 rows x 10 columns]
```

0.0.4 A Survey of ID list and Label list

```
In [14]: labels=readFile.label
         ids=readFile.id
         lines=len(readFile)
         idList=[]
         labelList=[]
         for i in range(lines):
             if ids[i] in idList:
                 continue
             else:
                 idList.append(ids[i])
                 labelList.append(labels[i])
         print(idList)
         print(len(idList))
         print(labelList)
```

```
[0, 1, 2, 3, 4, 6, 5, 7, 8, 9, 12, 11, 10, 13, 14, 16, 15]
17
['Car', 'Car', 'Two-Wheeler', 'Pedestrian', 'Pedestrian', 'Pedestrian', 'Pedestrian', 'Pedestri
```

1 Benchmarking

`totalFrames` stores the total number of the frames that we have for processing

`totalID` Stores the total number of distinct objects we have for processing

`videoNo` Stores the total number of annotated sequences or videos we have

`totalbbox` gives the total number of bounding boxes we have in the entire dataset

`totalPeds` gives the total number of annotated bounding boxes occluded as well as unoccluded

`occPeds` gives the total number of occluded annotated pedestrian bounding boxes

Similar data is plotted for Bus, Two wheelers, cars, Trucks, Autos, Cyclists and Animals respectively

`irrcount` represents the number of bounding boxes of annotated road signs

`ijkPedCross` `ijkStop` are lists which stores the video folder of annotators, the corresponding video and frames in which the road signs data has been annotated.

```
In [15]: def height(ybr,ytl):
             return abs(ybr-ytl)
         def width(xbr,xtl):
             return abs(xbr-xtl)
         def avg(arr,total):
             npArr=np.asarray(arr)
             average=np.sum(npArr)/total
             return average

In [16]: totalFrames=0
         totalID=0
         videoNo=0
         totalbbox=0
         totalPeds=0
         occPeds=0
         totalBus=0
         occBus=0
         totalTwo=0
         occTwo=0
         totalCar=0
         occCar=0
         totalTruck=0
         occTruck=0
         totalAuto=0
         occAuto=0
         totalCyclist=0
         occCyclist=0
         totalAnimal=0
         occAnimal=0
         allClasses=['Car', 'Two-Wheeler', 'Pedestrian', 'Auto-Rickshaw', 'Truck', 'Cyclist',
         irrcount=0
         ijkPedCross=[]
         ijkPark=[]
         ijkStop=[]
         ijkRight=[]
         ijkU=[]
         pedWidth=[]
         pedHeight=[]
         carWidth=[]
         carHeight=[]
         busWidth=[]
         busHeight=[]
         autoWidth=[]
         autoHeight=[]
         truckWidth=[]
         truckHeight=[]
         twoWidth=[]
         twoHeight=[]
```

```
cyclistWidth=[]
cyclistHeight=[]
animalWidth=[]
animalHeight=[]
uniqPeds=0
uniqCars=0
uniqTrucks=0
uniqBuses=0
uniqCyclists=0
uniqTwos=0
uniqAnimals=0
uniqAutos=0
#i represents the outer folders representing different persons' video folders
for i in range(1,20):
    inpath=path+"/"+str(i)
    for j in range(1,8):
        try:
            fullPath=inpath+"/"+str(j)+"/00"+str(j)+".csv"#j loop represents the name
            readFile=pd.read_csv(fullPath)
            totalbbox=totalbbox+len(readFile)
            pedsIDuniq=[]
            carsIDuniq=[]
            busesIDuniq=[]
            trucksIDuniq=[]
            cyclistsIDuniq=[]
            twosIDuniq=[]
            animalsIDuniq=[]
            autosIDuniq=[]
            totalFrames=totalFrames + max(readFile.frameNumber)#total frames added
            ids=readFile.id
            idSet=set(ids)
            totalID=totalID+len(idSet)#total number of dstinct ids or total number of
            videoNo=videoNo+1#total number of videos
            for k in range(len(readFile)):
                if(readFile.label[k]=='Pedestrian'):
                    totalPeds=totalPeds+1
                    pedWidth.append(width(readFile.x_BR[k],readFile.x_TL[k]))
                    pedHeight.append(height(readFile.y_BR[k],readFile.y_TL[k]))
                    if(not(readFile.id[k] in pedsIDuniq)):
                        uniqPeds=uniqPeds+1
                        pedsIDuniq.append(readFile.id[k])
                    if(readFile.isOccluded[k]==1):
                        occPeds=occPeds+1
                elif(readFile.label[k]=='Car'):
                    totalCar=totalCar+1
                    carWidth.append(width(readFile.x_BR[k],readFile.x_TL[k]))
                    carHeight.append(height(readFile.y_BR[k],readFile.y_TL[k]))
                    if(not(readFile.id[k] in carsIDuniq)):
```

```
            uniqCars=uniqCars+1
            carsIDuniq.append(readFile.id[k])
        if(readFile.isOccluded[k]==1):
            occCar=occCar+1
    elif(readFile.label[k]=='Truck'):
        totalTruck=totalTruck+1
        truckWidth.append(width(readFile.x_BR[k],readFile.x_TL[k]))
        truckHeight.append(height(readFile.y_BR[k],readFile.y_TL[k]))
        if(not(readFile.id[k] in trucksIDuniq)):
            uniqTrucks=uniqTrucks+1
            trucksIDuniq.append(readFile.id[k])
        if(readFile.isOccluded[k]==1):
            occTruck=occTruck+1
    elif(readFile.label[k]=='Bus'):
        totalBus=totalBus+1
        busWidth.append(width(readFile.x_BR[k],readFile.x_TL[k]))
        busHeight.append(height(readFile.y_BR[k],readFile.y_TL[k]))
        if(not(readFile.id[k] in busesIDuniq)):
            uniqBuses=uniqBuses+1
            busesIDuniq.append(readFile.id[k])
        if(readFile.isOccluded[k]==1):
            occBus=occBus+1
    elif(readFile.label[k]=='Two-Wheeler'):
        totalTwo=totalTwo+1
        twoWidth.append(width(readFile.x_BR[k],readFile.x_TL[k]))
        twoHeight.append(height(readFile.y_BR[k],readFile.y_TL[k]))
        if(not(readFile.id[k] in twosIDuniq)):
            uniqTwos=uniqTwos+1
            twosIDuniq.append(readFile.id[k])
        if(readFile.isOccluded[k]==1):
            occTwo=occTwo+1
    elif(readFile.label[k]=='Auto-Rickshaw'):
        totalAuto=totalAuto+1
        autoWidth.append(width(readFile.x_BR[k],readFile.x_TL[k]))
        autoHeight.append(height(readFile.y_BR[k],readFile.y_TL[k]))
        if(not(readFile.id[k] in autosIDuniq)):
            uniqAutos=uniqAutos+1
            autosIDuniq.append(readFile.id[k])
        if(readFile.isOccluded[k]==1):
            occAuto=occAuto+1
    elif(readFile.label[k]=='Cyclist'):
        totalCyclist=totalCyclist+1
        cyclistWidth.append(width(readFile.x_BR[k],readFile.x_TL[k]))
        cyclistHeight.append(height(readFile.y_BR[k],readFile.y_TL[k]))
        if(not(readFile.id[k] in cyclistsIDuniq)):
            uniqCyclists=uniqCyclists+1
            cyclistsIDuniq.append(readFile.id[k])
        if(readFile.isOccluded[k]==1):
```

```
                        occCyclist=occCyclist+1
                    elif(readFile.label[k]=='Animal'):
                        totalAnimal=totalAnimal+1
                        animalWidth.append(width(readFile.x_BR[k],readFile.x_TL[k]))
                        animalHeight.append(height(readFile.y_BR[k],readFile.y_TL[k]))
                        if(not(readFile.id[k] in animalsIDuniq)):
                            uniqAnimals=uniqAnimals+1
                            animalsIDuniq.append(readFile.id[k])
                        if(readFile.isOccluded[k]==1):
                            occAnimal=occAnimal+1
                    elif(readFile.label[k]=='Pedestrian Crossing'):
                        irrcount=irrcount+1
                        ijkPedCross.append(str(i)+"-"+str(j)+"-"+str(k))
                    elif(readFile.label[k]=='No Parking'):
                        irrcount=irrcount+1
                        ijkPark.append(str(i)+"-"+str(j)+"-"+str(k))
                    elif(readFile.label[k]=='No Stopping'):
                        irrcount=irrcount+1
                        ijkStop.append(str(i)+"-"+str(j)+"-"+str(k))
                    elif(readFile.label[k]=='Right Turn'):
                        irrcount=irrcount+1
                        ijkRight.append(str(i)+"-"+str(j)+"-"+str(k))
                    elif(readFile.label[k]=='U-Turn'):
                        irrcount=irrcount+1
                        ijkU.append(str(i)+"-"+str(j)+"-"+str(k))
                    elif(not(readFile.label[k] in allClasses)):
                        allClasses.append(readFile.label[k])
                except:
                    print("file not found",i,j)
                    videoNo=videoNo-1#unfound videos are unadded from the total video number
```

```
file not found 1 1
file not found 3 7
file not found 4 4
file not found 4 5
file not found 4 6
file not found 14 5
file not found 14 6
file not found 14 7
file not found 15 3
file not found 15 4
file not found 15 5
file not found 15 6
file not found 15 7
file not found 16 5
file not found 16 6
file not found 16 7
file not found 17 5
```

```
file not found 17 6
file not found 17 7
file not found 18 5
file not found 18 6
file not found 18 7
file not found 19 5
file not found 19 6
file not found 19 7
```

In [17]:
```
print("Total number of frames",totalFrames)
print("Total number of distinct ids or total number of distinct objects",totalID)
print("Total sequences or videos",videoNo)
print("Total Number of Bounding Boxes", totalbbox)
print("Total Number of pedestrians", totalPeds )
print("occluded pedestrians",occPeds)
print("number of unique pedestrians",uniqPeds)
print("Total Number of cars", totalCar )
print("occluded cars",occCar)
print("number of unique Cars",uniqCars)
print("Total Number of buses", totalBus )
print("occluded buses",occBus)
print("number of unique buses",uniqBuses)
print("Total Number of Trucks", totalTruck )
print("occluded Trucks",occTruck)
print("number of unique trucks",uniqTrucks)
print("Total Number of Animals", totalAnimal )
print("occluded Animals",occAnimal)
print("number of unique animals",uniqAnimals)
print("Total Number of Cyclists", totalCyclist )
print("occluded Cyclists",occCyclist)
print("number of unique cyclists",uniqCyclists)
print("Total Number of two-wheelers", totalTwo )
print("occluded two-wheelers",occTwo)
print("number of unique two-wheelers",uniqTwos)
print("Total Number of Auto-Rikshaws", totalAuto )
print("occluded Auto-Rikshaws",occAuto)
print("number of unique autos",uniqAutos)
print("totalbbox", totalPeds+totalCar+totalBus+totalTruck+totalCyclist+totalTwo+totalA
print("Total number of bounding boxes for the 8 class objects", totalbbox-irrcount)
print("All the classes of data that are present",allClasses)
avgPedWidth=avg(pedWidth,totalPeds)
avgPedHeight=avg(pedHeight,totalPeds)
avgCarWidth=avg(carWidth,totalCar)
avgCarHeight=avg(carHeight,totalCar)
avgTruckWidth=avg(truckWidth,totalTruck)
avgTruckHeight=avg(truckHeight,totalTruck)
avgBusWidth=avg(busWidth,totalBus)
```

```
avgBusHeight=avg(busHeight,totalBus)
avgAnimalWidth=avg(animalWidth,totalAnimal)
avgAnimalHeight=avg(animalHeight,totalAnimal)
avgTwoWidth=avg(twoWidth,totalTwo)
avgTwoHeight=avg(twoHeight,totalTwo)
avgCyclistWidth=avg(pedWidth,totalCyclist)
avgCyclistHeight=avg(pedHeight,totalCyclist)
avgAutoWidth=avg(pedWidth,totalAuto)
avgAutoHeight=avg(pedHeight,totalAuto)
print("average pedestrian width",avgPedWidth)
print("average pedestrian Height",avgPedHeight)
print("average Car width",avgCarWidth)
print("average Car Height",avgCarHeight)
print("average Bus width",avgBusWidth)
print("average Bus Height",avgBusHeight)
print("average Auto width",avgAutoWidth)
print("average Auto Height",avgAutoHeight)
print("average Cyclist width",avgCyclistWidth)
print("average Cyclist Height",avgCyclistHeight)
print("average Two Wheeler width",avgTwoWidth)
print("average Two Wheeler Height",avgTwoHeight)
print("average Truck width",avgTruckWidth)
print("average Truck Height",avgTruckHeight)
print("average Animal width",avgAnimalWidth)
print("average Animal Height",avgAnimalHeight)
print("Number of road sign boxes",irrcount)
print("The person-video-frame info for annotation of Pedestrian Crossing marking \n",
print("The person-video-frame info for annotation of Parking Sign marking \n",ijkPark
print("The person-video-frame info for annotation of Stop Sign marking \n",ijkStop)
print("The person-video-frame info for annotation of Right Turn marking \n",ijkRight)
print("The person-video-frame info for annotation of U-Turn marking \n",ijkU)
```

```
Total number of frames 53797
Total number of distinct ids or total number of distinct objects 2493
Total sequences or videos 83
Total Number of Bounding Boxes 253468
Total Number of pedestrians 20036
occluded pedestrians 2515
number of unique pedestrians 371
Total Number of cars 143051
occluded cars 32944
number of unique Cars 1325
Total Number of buses 7244
occluded buses 2017
number of unique buses 44
Total Number of Trucks 10462
occluded Trucks 2511
number of unique trucks 79
```

```
Total Number of Animals 131
occluded Animals 0
number of unique animals 4
Total Number of Cyclists 3956
occluded Cyclists 743
number of unique cyclists 58
Total Number of two-wheelers 50907
occluded two-wheelers 5981
number of unique two-wheelers 467
Total Number of Auto-Rikshaws 17436
occluded Auto-Rikshaws 2699
number of unique autos 146
totalbbox 253223
Total number of bounding boxes for the 8 class objects 253223
All the classes of data that are present ['Car', 'Two-Wheeler', 'Pedestrian', 'Auto-Rickshaw',
average pedestrian width 28.7980135756
average pedestrian Height 77.5950788581
average Car width 90.54169492
average Car Height 68.7031338474
average Bus width 200.093318609
average Bus Height 153.474737714
average Auto width 33.0922803395
average Auto Height 89.1658063776
average Cyclist width 145.85364004
average Cyclist Height 392.996713852
average Two Wheeler width 49.0557880056
average Two Wheeler Height 78.7074272693
average Truck width 132.55400497
average Truck Height 118.213630281
average Animal width 40.1450381679
average Animal Height 42.7099236641
Number of road sign boxes 245
The person-video-frame info for annotation of Pedestrian Crossing marking
 ['5-7-2193', '5-7-2196', '5-7-2200', '5-7-2204', '5-7-2208', '5-7-2211', '5-7-2215', '5-7-221
The person-video-frame info for annotation of Parking Sign marking
 ['6-7-449', '6-7-454', '6-7-459', '6-7-464', '6-7-469', '6-7-474', '6-7-479', '6-7-484', '6-7
The person-video-frame info for annotation of Stop Sign marking
 ['8-7-793', '8-7-797', '8-7-799', '8-7-805', '8-7-806', '8-7-813', '8-7-817', '8-7-821', '8-7
The person-video-frame info for annotation of Right Turn marking
 ['9-7-46', '9-7-52', '9-7-54', '9-7-59', '9-7-64', '9-7-69', '9-7-74', '9-7-79', '9-7-84', '9
The person-video-frame info for annotation of U-Turn marking
 ['9-7-931', '9-7-934', '9-7-940', '9-7-946', '9-7-952', '9-7-958', '9-7-963', '9-7-968', '9-7
```

In [ ]: