Assignment 4
CE 787A: Computational Tools for Transportation Engineering
Total Marks: 35

Instructions: Use Jupyter notebook to work on this assignment using Python language. All cell outputs and visualizations (if any) should be visible and necessary comments should be put to make the code readable. Once the code is ready, convert it to HTML (File > Download as HTML), save the html as pdf, and submit the pdf. The file name should be AssignmentNumber_RollNo_FirstName.pdf (only one pdf for the assignment)
Also, submit the *.ipynb file* of the jupyter notebook separately.

1. The zip file "ques1_data.zip" contains 2 images "car_tent.jpg" and "dog.jpg". The zip file also contains "ques1_annotation.csv", which contains position of different objects present in the two images. The schema of the csv file is *filename, class, x, y, width, height*

- filename: name of the jpg file
- class: object type
- x: x-coordinate of the top left-corner of the object (see Figure 1)
- y: y-coordinate of the top-left corner of the object (see Figure 1)
- width: width of the object (see Figure 1)
- height: height of the object (see Figure 1)

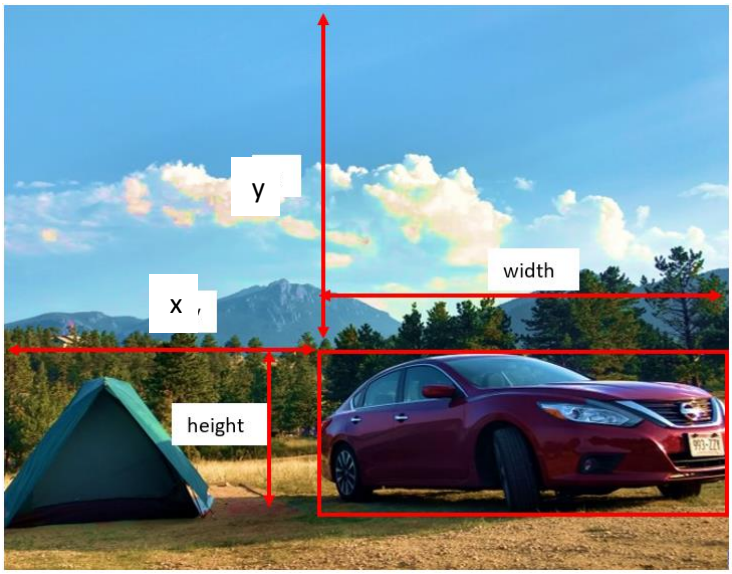

Figure 1. Sample position of object in image

With this required information, perform the following tasks:

a) Read the annotation csv file. How many unique classes of object are present in the two images? 2

b) Extract all objects present in the two images based on the position of the objects provided in the annotation file. Save all object images that you have extracted. Do not extract objects manually. You should be using "for" loop or such for extracting and saving all objects. You can use any naming convention for the filenames of the extracted objects. 6+4

c) Display the "tent" object from "car_tent.jpg" and "bicycle" object from "dog.jpg" in your jupyter notebook. 3

2. The "*ques2_data.zip*" contains two csv files: "*threshold_data.csv*" and "*segment_data.csv*".

The schema of segment_data.csv is: *Code, Road, Direction, Mileage*
Code: Unique sensor ID
Road: Road name where the sensor is located
Direction: Road direction where sensor is located (EB means Eastbound, WB means Westbound).
Mileage: Mileage of the road where the sensor is located. This helps to provide spatial location of the sensor.

The schema of *threshold_data.csv* is: *Code, Time, Speed*
Code: Unique sensor ID
Time: Represents 15-minute time interval period
Speed: Expected threshold speed for the given sensor for the given time period in miles per hour (mph)

The *Time* column is obtained after aggregating timestamps into 15-minute interval period. Therefore, period can be 0/1/2/3 (there are four 15-minute interval in an hour). Then, Time column is obtained using: $Time = Hour \times 100 + Period$. Therefore,
800 represents 08:00:00-08:14:00, 801 represents 08:15:00-08:29:00,
1603 represents 16:45:00-16:59:00 and so on.
In this assignment, you don't need to change the *Time* column to any other format.

With the above information, perform the following tasks:

a) Read both the data files "*threshold_data.csv*" and "*segment_data.csv*". Merge the two files based on the "Code" column. You will use this merged dataframe for the remaining question. 3

b) Convert any values of threshold *speed* greater than 45 mph to 45 mph and any values of threshold speed less than 0 mph to 0 mph i.e., the modified maximum and minimum value of threshold speed in the merged dataframe will be 45 mph and 0 mph respectively. 3

c) Filter the merged dataframe for *Road* = 'I-235' and *Direction* = 'WB' 2

d) Use only 3 columns '*Distance'*, '*Time'*, and '*Speed'* and then pivot table such that the columns represent '*Time'*, rows indicate '*Distance'*, the values represent '*Speed'*. Use '*Distance'* as index column and this pivot dataframe should be sorted based on the distance. Sample output is shown below: 4

hourper	0	1	2	3	...	2301	2302	2303
Distance								
0.000000	43.75	43.0	45.00	45.0	...	32.0	45.0	45.0
0.676521	45.00	45.0	45.00	45.0	...	45.0	45.0	45.0
1.741511	40.00	45.0	45.00	45.0	...	45.0	45.0	45.0
2.056675	45.00	45.0	45.00	45.0	...	41.0	45.0	45.0

e) Convert this pivot dataframe into a matrix (Hint: You can use *df*.values). Plot this matrix as an image and show the image in your jupyter notebook. 3

f) Apply medianBlur filter to this matrix to denoise the image with kernel window size 3. Note, you need to convert the matrix to '*float32'* type before applying the median filter. Plot this denoise matrix as an image and show in jupyter notebook. Comment whether applying filter has been able to remove noise in the image or not. 5