Traffic density estimation Group projec t

- Anjali Raj 20EE10010

Objective

Measure Dynamic traffic density from a video of traffic intersection and analyse the plots.

The Dynamic density could provide useful information about the traffic which can be used by traffic signals for avoiding traffic-jam.

SUB-PARTS PERFORMED

- Background subtraction
- Camera angle and image colour correction
- Frame cropping
- Vehicle count
- Pixel density plotting
- Car quantity plotting

Github-repo-link:

https://github.com/IshanManchanda/diy-traffic-density-estimation/tree/master/image_processing

Procedure

The project has been divided into 3 subtasks, for better understanding and implementation.

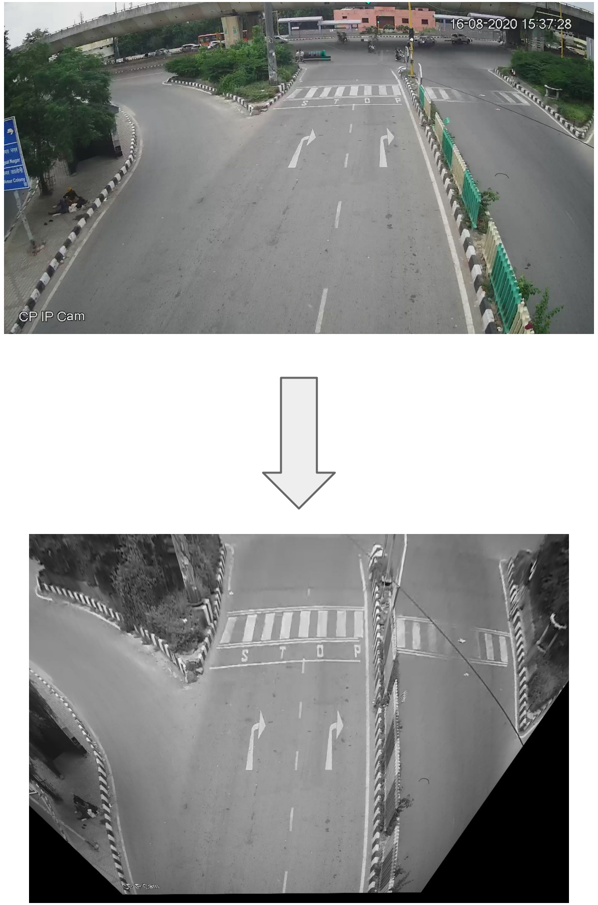
1. Image colour and angle correction
2. Calculating densities from the video by applying subtask 1 to each frame of the video.
3. Plotting the pixel density along with the car density and analysing the two graphs.

SUBTASK EXPLAINED - SUBTASK 1

Every Camera installed at signals has a skewed view of the road in front of it, to maximise coverage.

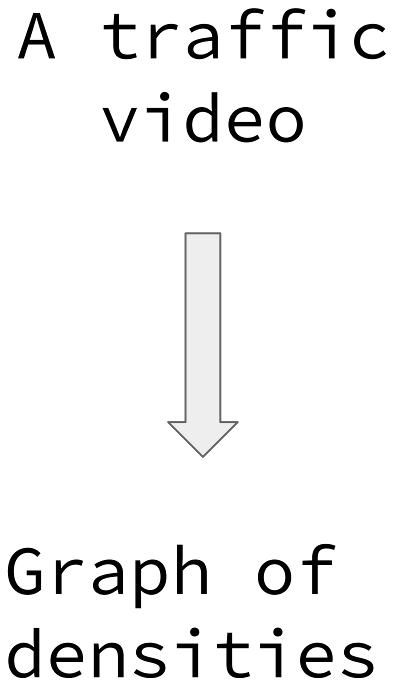
This leads to closer objects being magnified and distorted. This distortion needs to be corrected before analysis of video, using certain transformations.

Also, the density estimation does not require colored video, and hence it needs to be converted to grayscale.



SUBTASK EXPLAINED - SUBTASK 2 & 3

- From the transformed video, estimations of both kinds of densities will be extracted with multiple optimisations.



A traffic video

Graph of densities

EXECUTION

video_write.py:CAMERA ANGLE CORRECTION ON WHOLE VIDEO

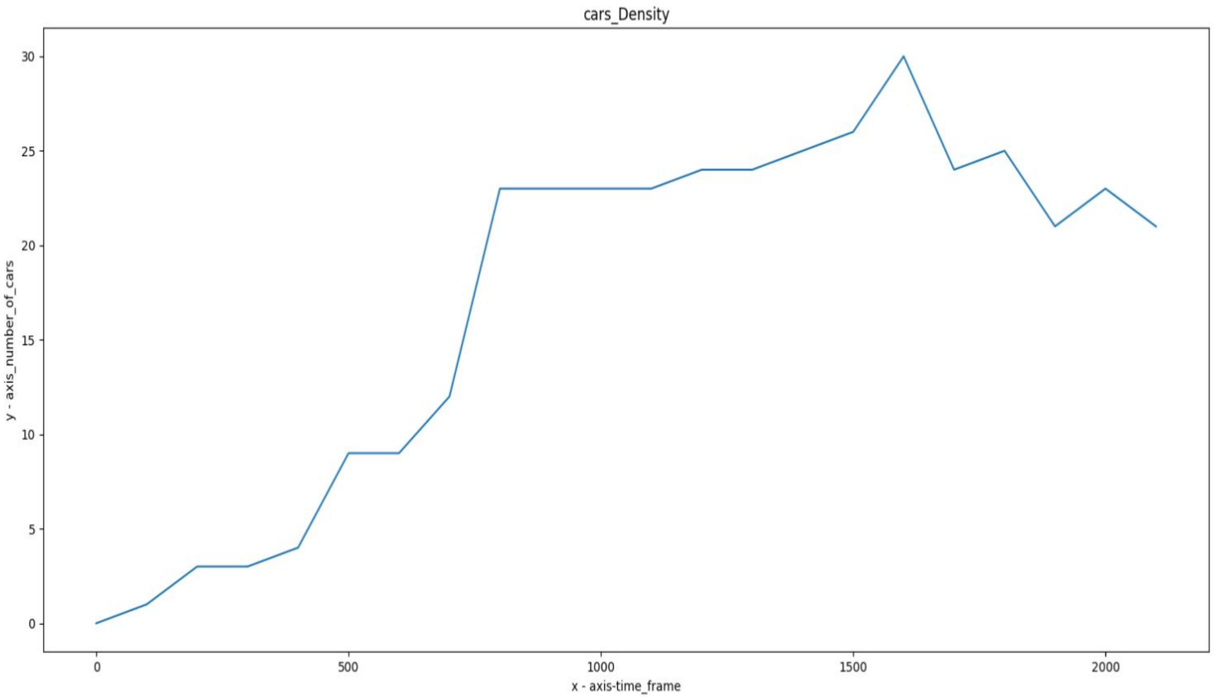
vehicle_sub.py:Subtracting the vehicles crossing the upper bar

vehicle_add.py:Adding the vehicles crossing the lower bar

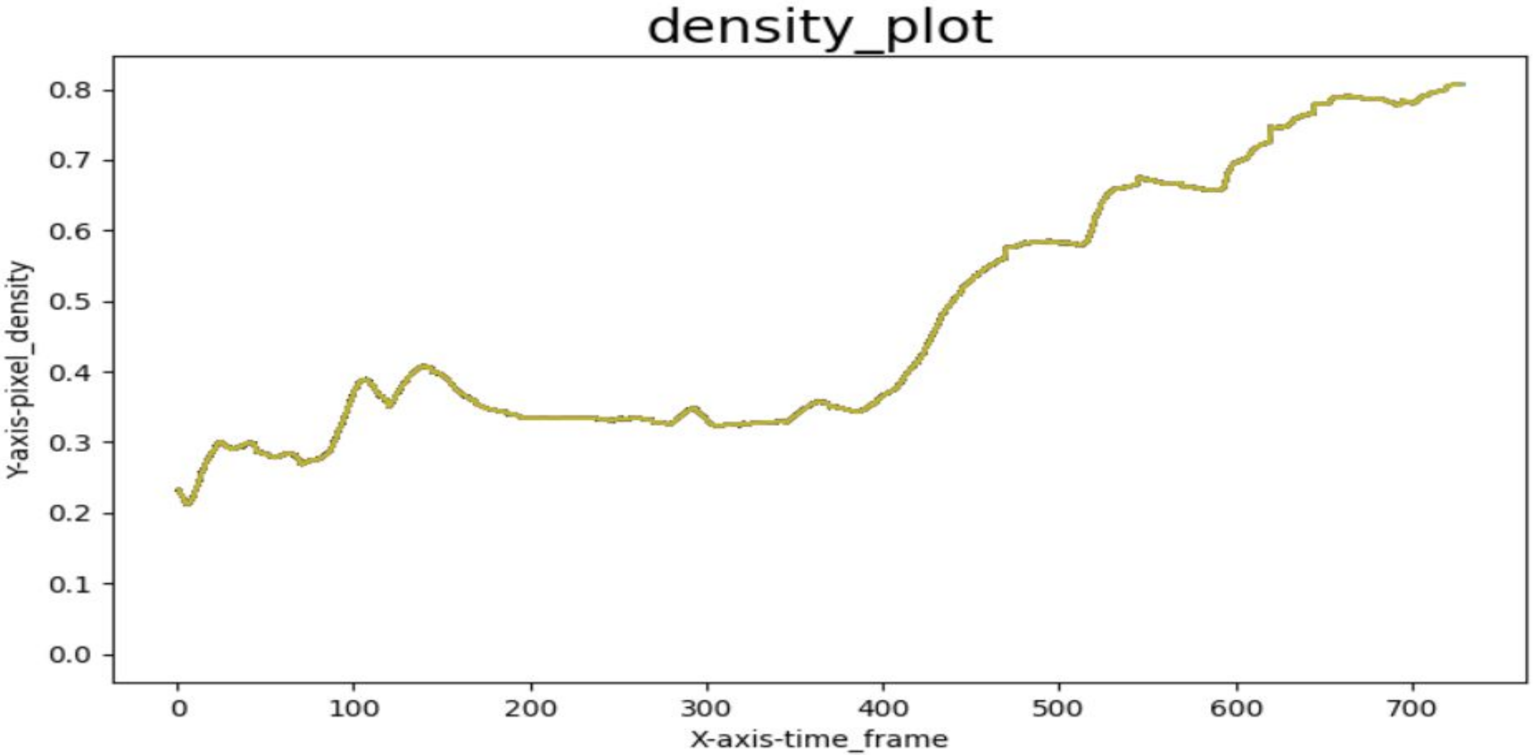
Car_plotter.py:plotting the current number of cars every 100 iterations.

Pixel_density_plotter.py:for plotting the pixel density of the cars.

OUTPUTS


cars_Density
30
25
20
15
10
5
0
y - axis_number_of_cars
0
500
1000
1500
2000
x - axis-time_frame

OUTPUTS


density_plot
Y-axis-pixel_density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
X-axis-time_frame
0
100
200
300
400
500
600
700

Scope of Improvement

- reduce resolution for each frame. Lower resolution frames might be processed faster, but have higher errors. Parameter can be resolution XxY.
- split work spatially across threads (application level pthreads) by giving each thread part of a frame to process. Parameter can be number of splits i.e. number of threads, if each thread gets one split. You might need to take care of boundary pixels that different threads process for utility.