**Chapter 5**

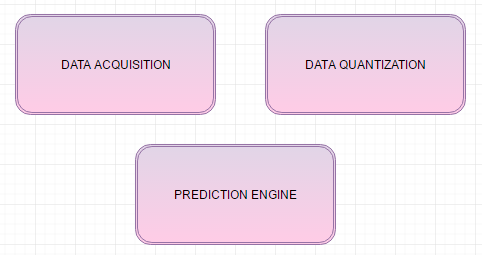
**SYSTEM DESIGN**

The proposed software system can be realized into 3 modules as below.

1. Data Acquisition Unit

2. Data Quantization Unit

3. Prediction Engine



**Figure 5.1: Modules of the software**

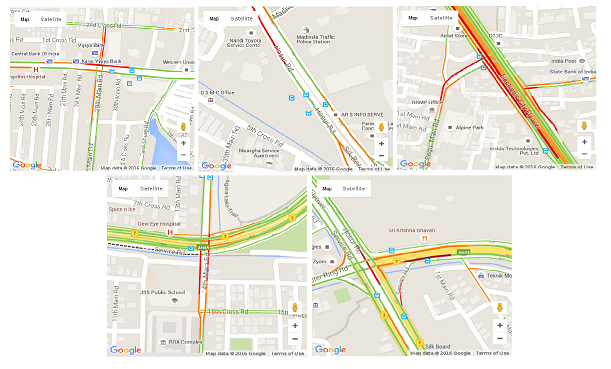
**5.1 Data Acquisition**

This is the sense organ of the entire software system. Data the required for the entire processing is obtained in this unit. Google map provides the traffic density of a given road at any given point of time. This is a real time unit, responsible for collecting the traffic data provided by the traffic layer of the google maps. The snapshots of the junctions under observation are taken periodically and stored for further processing.

A headless browser named PhantomJS pings the traffic layer of the google maps periodically and the snapshots are captured. This is a data critical software in which the snapshots of the traffic layer plays an important role.

**5.1.1 GetSnap**

We calculate the percentage of traffic density by considering the images retrieved from Google maps using phantomjs as shown in Figure 5.2.



**Figure 5.2: Snapshot taken at 5 (axatech, hosur-road, electronic-city, ringroad, silkboard) junctions**

In above snapshots(Figure 5.2), red line represents high density traffic, yellow line represents moderate and green line represents low density traffic. Images are taken for every one minute in all the junctions considered.

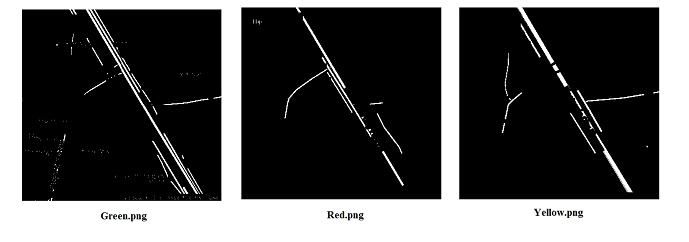
To get these snapshots, the code snapshot.js is run which includes map.html which accepts latitude and longitude of the junctions to be considered.

**5.2 Data Quantization**

The Data obtained by the previous unit is the imagic data. Reverse engineering is followed to obtain the quantitative data. The snapshots obtained contain colours that distinguish between the traffic densities. Red represent heavy traffic density, Green represent low traffic density and so on, the filter algorithm provided by the Open source Computer Vision library (OpenCV) is applied on to the coloured images and the corresponding binary images with black and white pixels are obtained. These images are further processed to obtain the quantitative data out of it, by using pixel count method. Percentage of the density values are calculated and stored for further processing.

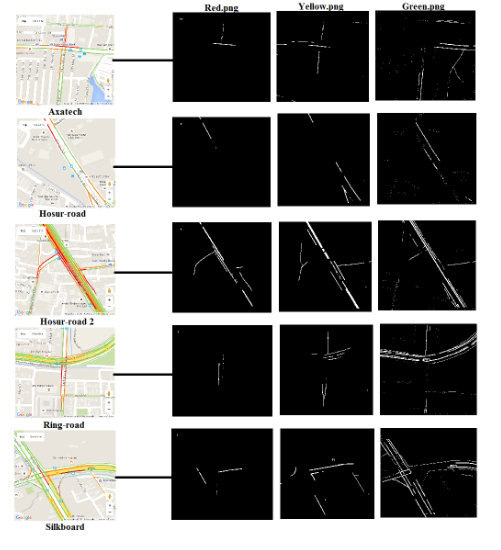
**5.2.1 Image Conversion**

To quantify the data, the coloured image is converted to binary image (black and white image) using OpenCV.



**Figure 5.3: Binary images generated by the filter program**

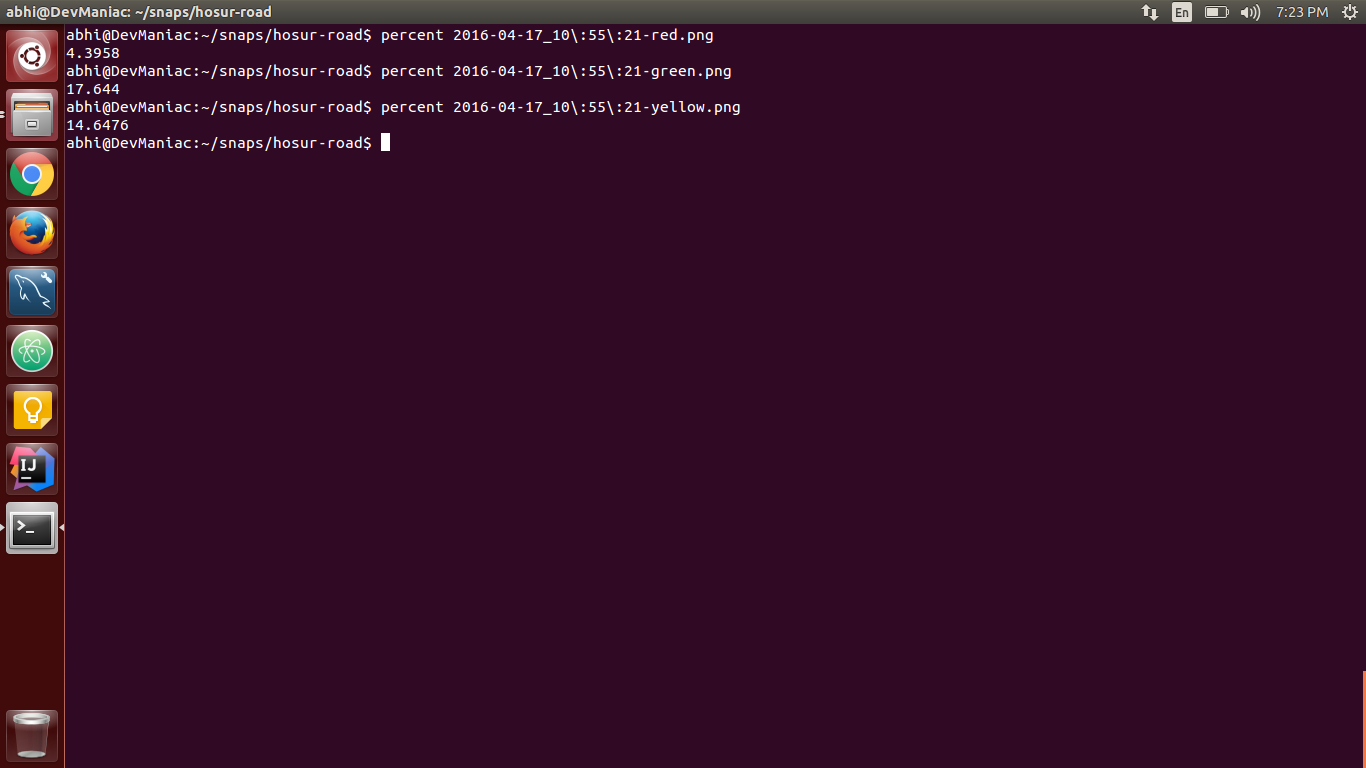
Red.png indicates high density traffic which is represented by white pixels. Yellow.png indicates moderate traffic and green.png indicates low traffic density which is represented by white pixels.

****

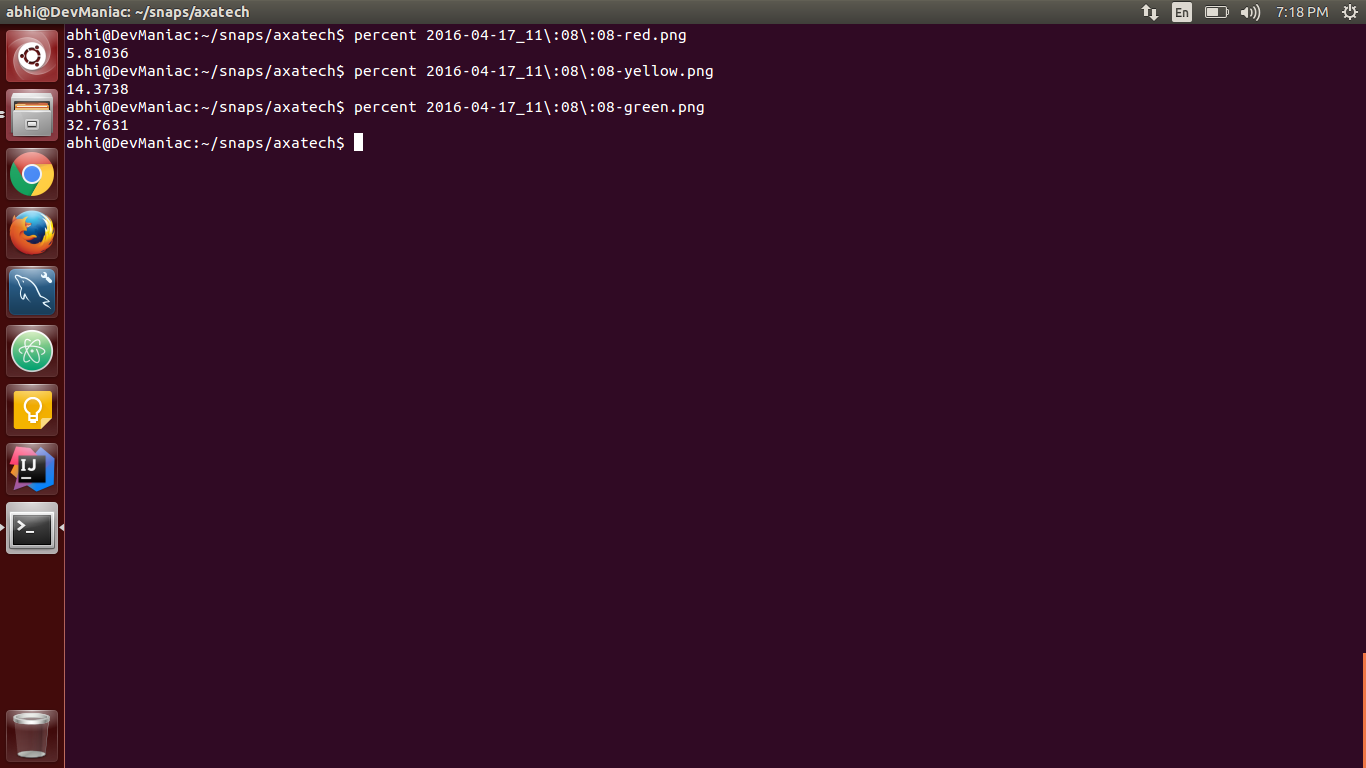
**Figure 5.4: Snapshot with filter result**

**5.2.2 Percentage Calculation**

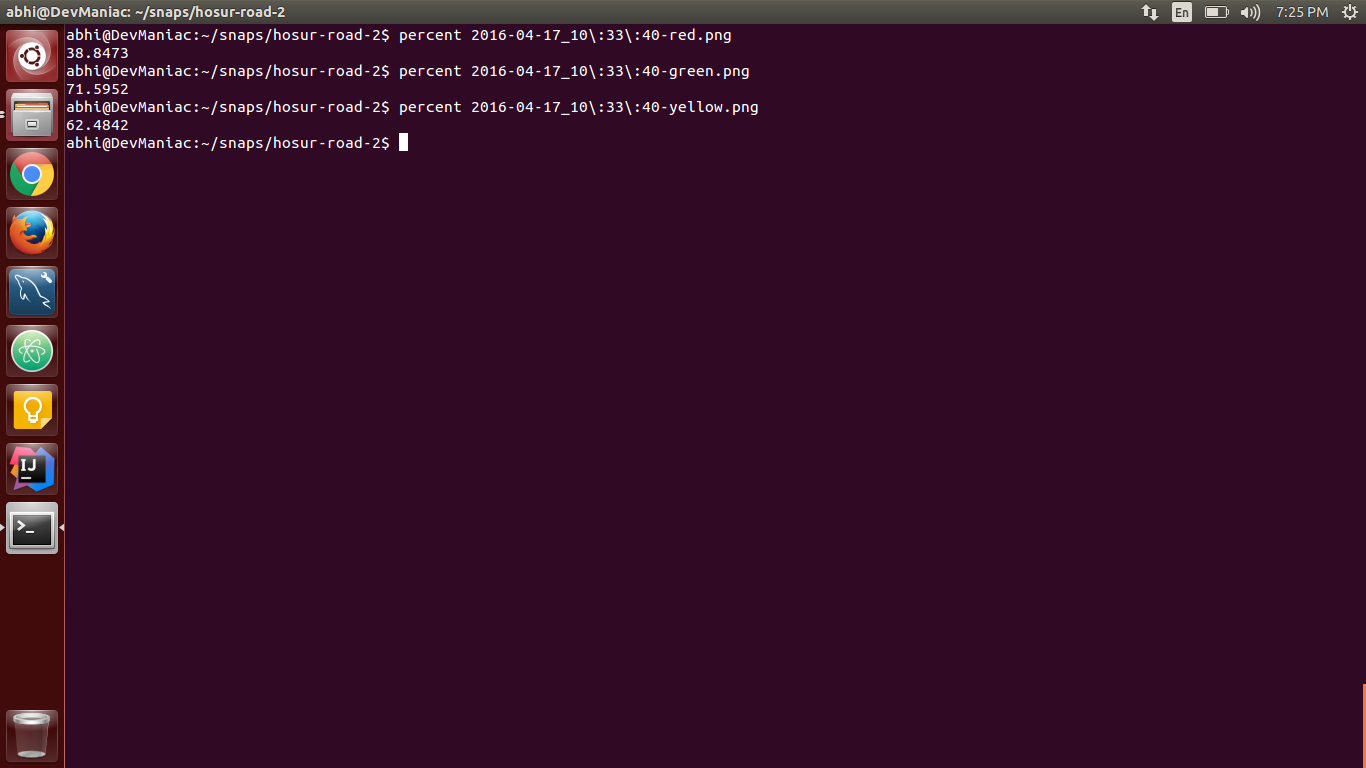
Binary image is given as input for percentage calculation. We take black and white pixel count for quantization and then we calculate white pixel percentage over total pixel count.

****

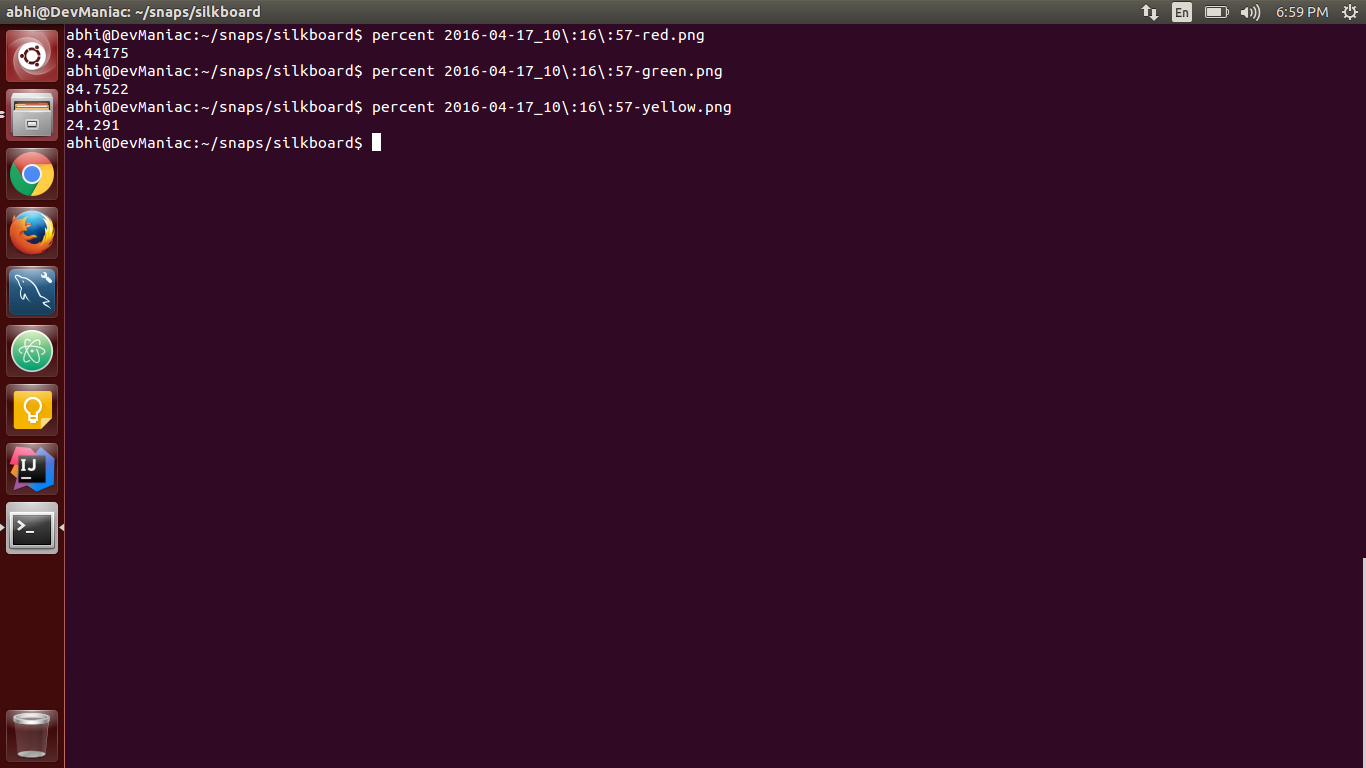
**Figure 5.5: Percentage calculation of the traffic density in hosur junction**

****

**Figure 5.6: Percentage calculation of the traffic density in axatech junction**

****

**Figure 5.7: Percentage calculation of the traffic density in hosur road 2 junction**

****

**Figure 5.8: Percentage calculation of the traffic density in silkboard junction**

**5.3 Prediction Engine**

Prediction Engine is responsible for forecasting few years ahead of the current traffic conditions. This unit clearly operates on the data that is provided by the previous two units discussed. The quantitative data of the traffic density is plotted against the time at which the snapshots were taken and a curve is fitted for the data of a single day. The same is followed for all the days on which the snaps were captured. The graph plotted for all the days are normalized to obtain an equation, using which the future traffic conditions are predicted.

**5.3.1 Plotting Graph**

Percentage values that are calculated using black and white pixels count are stored in the database. These values are then fetched from the database to plot the graph and extrapolate the graph using the GSL. By extrapolating the graph, we can learn the near future traffic rate.

**Figure 5.9: Graph plotted based on the traffic densities of the axatech junction calculated and the time at which the snaps were taken**

These graphs plotted using google charts are normalized and equations obtained from it using GNU scientific library is used to forecast the traffic few years down the line.

**5.4 Summary**

This chapter talks about the system design of the application. Brief explanation of the modules that comprises the application and its working is given.