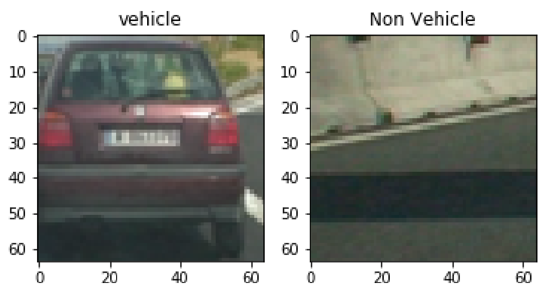
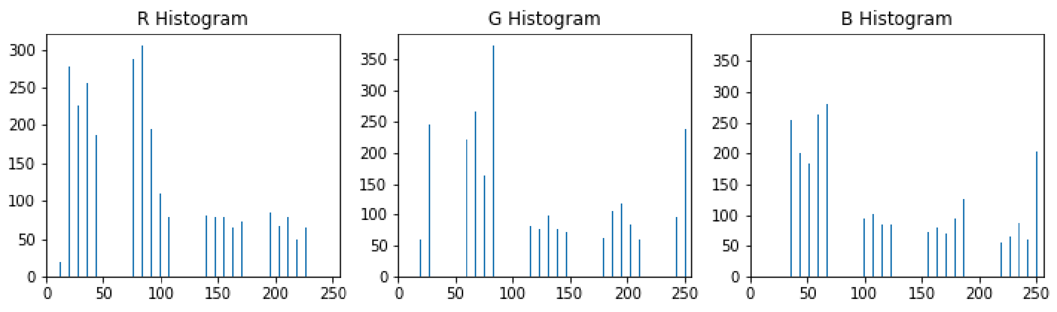
Vehicle Detection Project

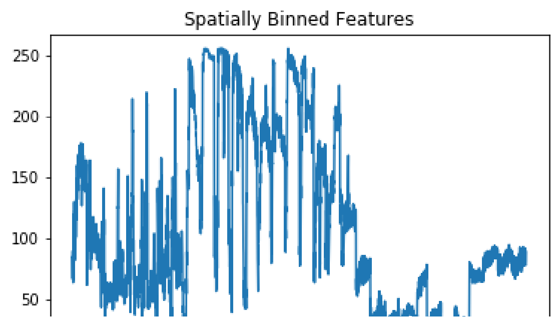
The goals / steps of this project are the following:

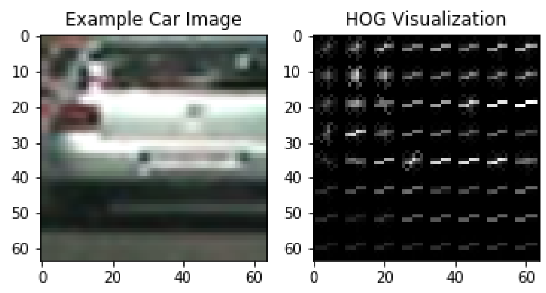
1.I was provided with a vehicle vs non-vehicle datasets, hence I choose to use all the images for prediction.



2.
Here we used two techniques to extract out the information out of the image.
First we used HOG or Histogram of Gradient and second we used Spatial Bining of Color. We combined both techniques to get the best results. The histogram of gradient function comes in built with sklearn where you can chose among various setting like normalization over a cell, how much pixels to choose and how much bins of histogram do we reqire.
In case of Spatially Binned Features we resize the image and flatten it.
Where HOG looks for gradient in colour and shape.







3. I used SVM or Support Vector machines to process these combined feature and created labels using np.ones for the correspoding values. I also normalized the features using Standard scalar from sklearn. Using this, I got around 98% accuracy.

4. Sliding Search's main algorithm is selecting two x1,y1 and x2,y2 values which can represent a rectangle. This rectangle is feed into the SVM for prediction. If the result is True or 1, we append the boxes/rectangle. There can be window overlap as objects tends to be not regularly spaced in real world.
There are four different windows used. One with 64x64,80x80,96x96 and 128x128 size.
They cover cars present in the bottom half of the image with varing degree of sizes. To detect cars of different sizes these sizes vary as the car move from near to far.
The overlap if from 0.15 ,0.2 ,0.3 and 0.5 in the windows respectively.

```
windows1 = slide_window(image, x_start_stop=[0, 1280], y_start_stop=[400,464],
                    xy_window=(64,64), xy_overlap=(0.15, 0.15))

windows2 = slide_window(image, x_start_stop=[0, 1280], y_start_stop=[400,480],

                    xy_window=(80,80), xy_overlap=(0.2, 0.2))

windows3 = slide_window(image, x_start_stop=[0, 1280], y_start_stop=[400,612],
                    xy_window=(96,96), xy_overlap=(0.3, 0.3))

windows4 = slide_window(image, x_start_stop=[0, 1280], y_start_stop=[400,660],
                    xy_window=(128,128), xy_overlap=(0.5, 0.5))
```

5.Sliding Search does result in many ovelapping windows which are removed if they represent the same object with the help of heatmap, where heatmap tends to represent the overall pixel value. If it's less than the threshold we delete the image as a false positive otherwise we use `scipy.ndimage.measurements.label()` which helps us to choose the box values.

Discussion includes some consideration of problems/issues faced, what could be improved about their algorithm/pipeline, and what hypothetical cases would cause their pipeline to fail.

I believe understand the way label works was an issue. The pipeline is straight forward. Pipeline might fail to identify cars in rain or shadow or in tunnels. Its not that robust.