Vehicle Detection and Tracking

TRAINING AND TEST DATA

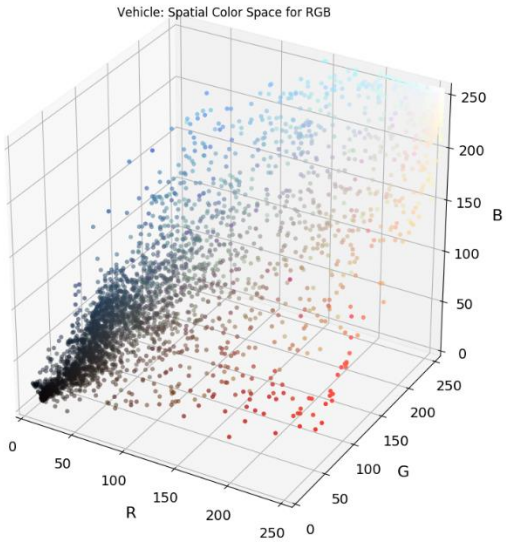
In this section I will mention details about the steps followed to update the features used to train and test the model.

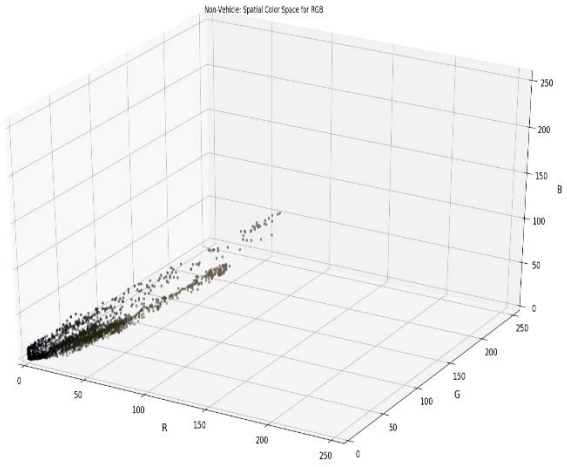
I have used the data project and I checked their length.

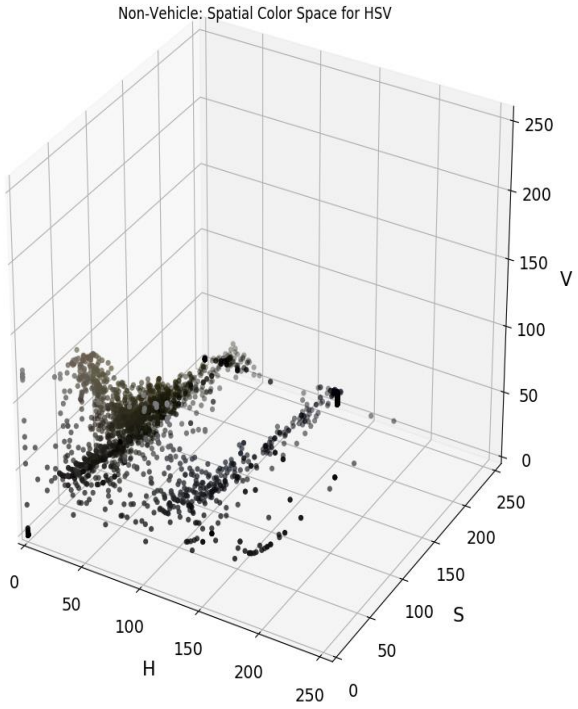
So I have found the size of Vehicle is 8792 and the non vehicle is 8968 so they are roughly the same.

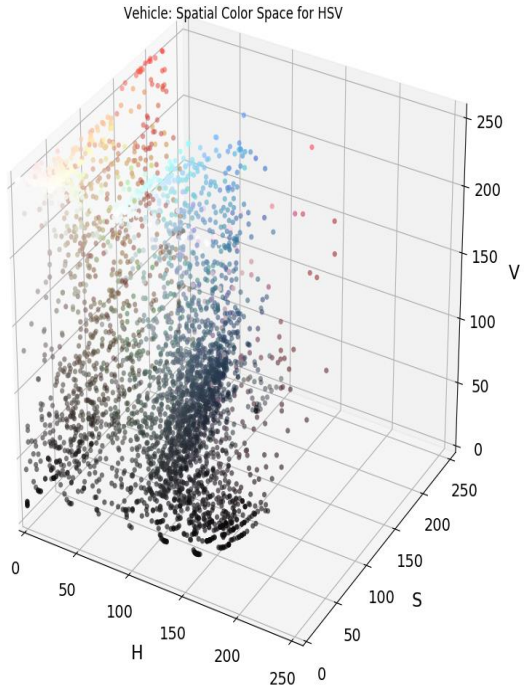
Spatial Binning Features

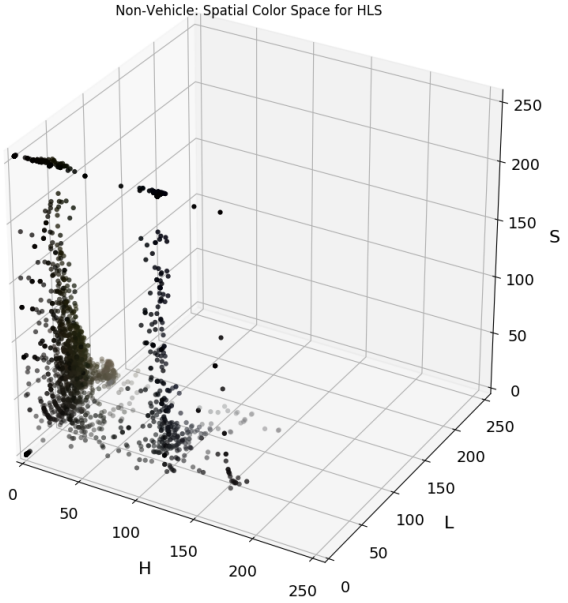
I plotted the different spatial distribution and I have selected YUV find below different color comparison between Vehicle and Non Vehicle

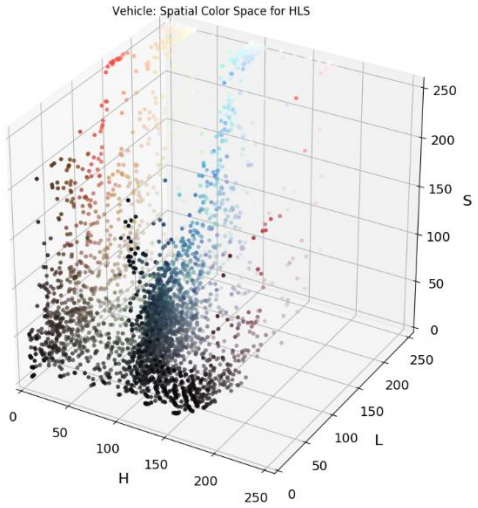


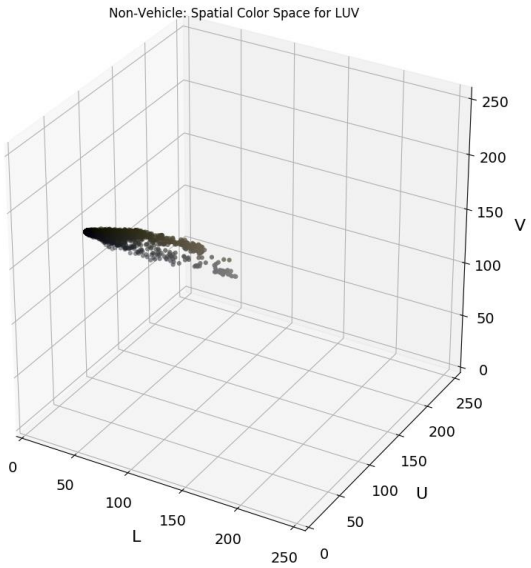


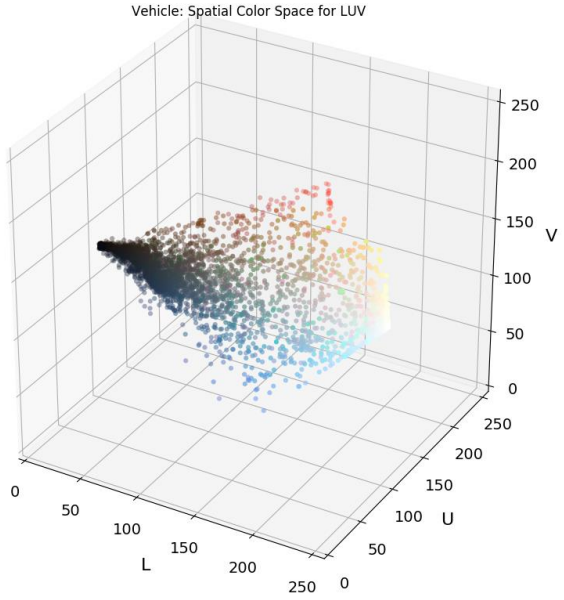

Non-Vehicle: Spatial Color Space for HSV
H
S
V

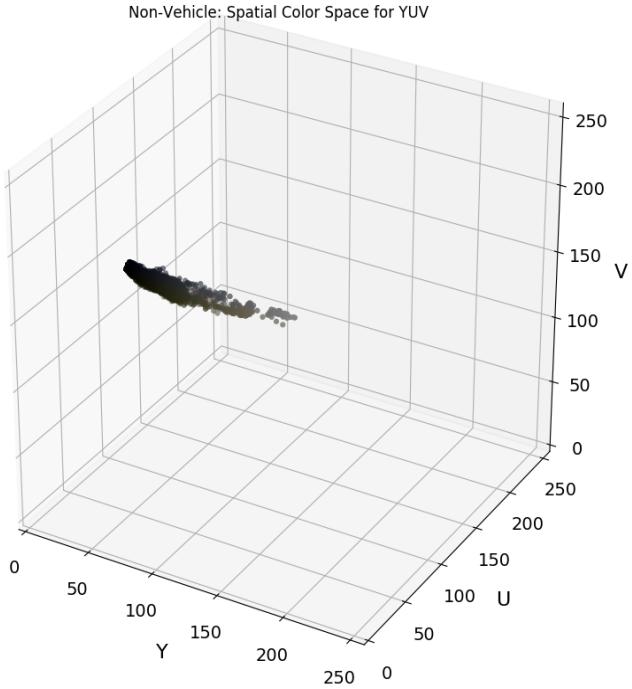

Vehicle: Spatial Color Space for HSV
H
S
V

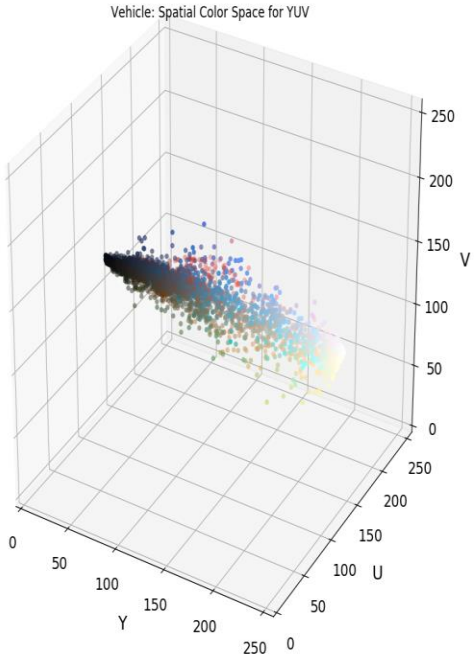

Non-Vehicle: Spatial Color Space for HLS
H
L
S


Vehicle: Spatial Color Space for HLS
H
L
S

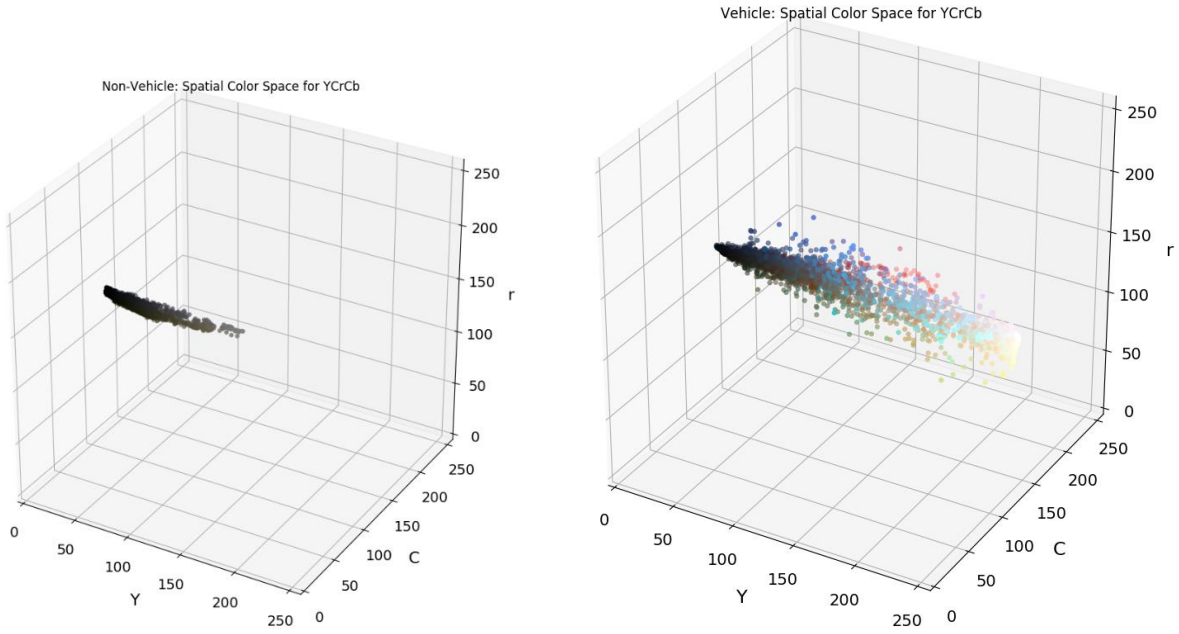








YUV and LUV are the best so I will use YUV.

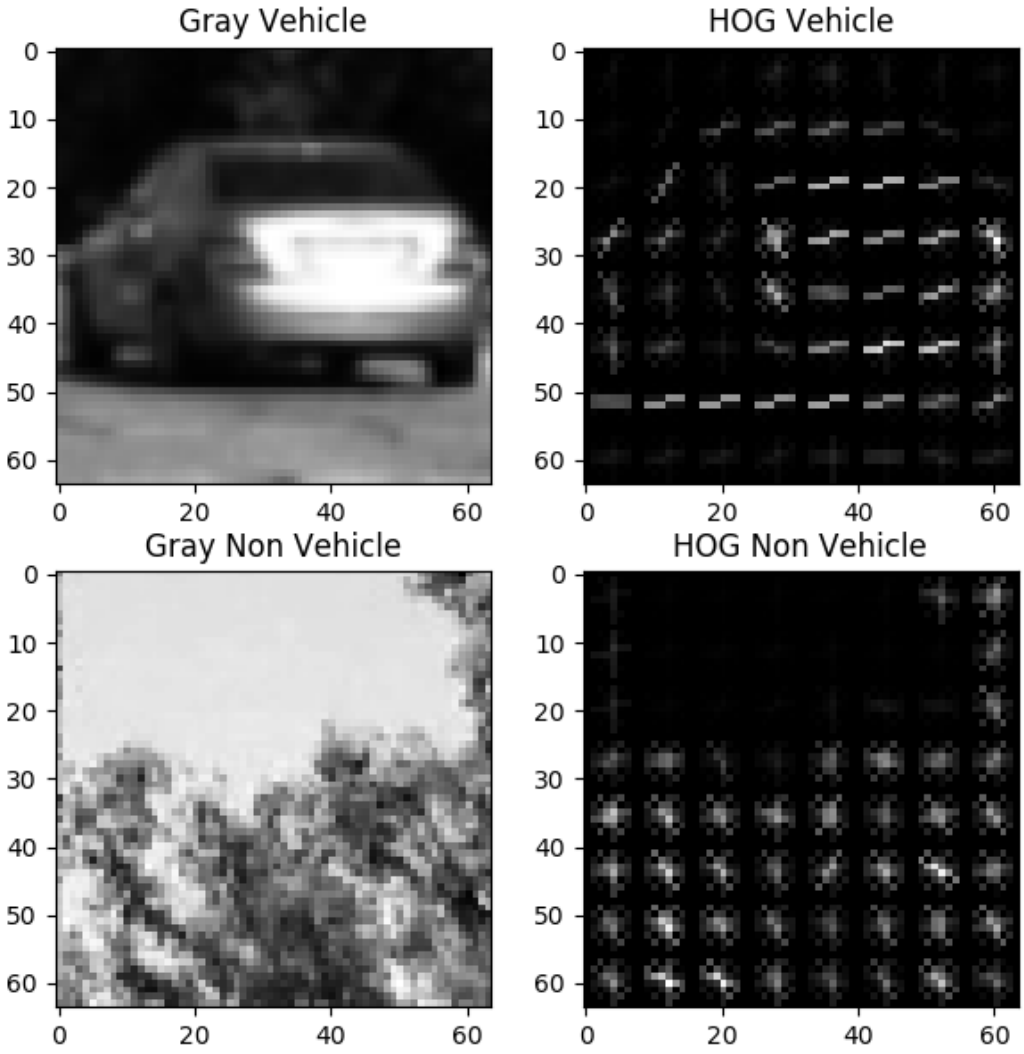


Histogram of Oriented Gradients Feature

Another feature we will use is HOG. Below an image to illustrate it.

HOG parameters

- orient = 9
- pix_per_cell = 8
- cell_per_block = 8



Combine and Prepare Train and Test Set

After I have extracted features from the data set I need to prepare the train set and test set:

1. Shuffle Data
2. Split it to Train set and test set
3. Label my set

SVM Classifier

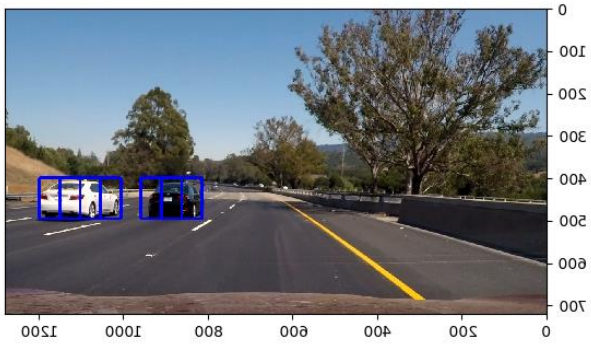
In the project I am using the SVM Classifier, I did not experiment any other classifiers as I see this one is robust Enough

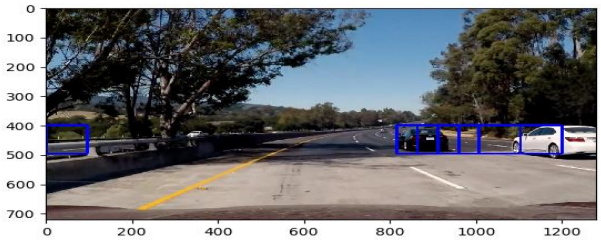
VEHICLE DETECTION

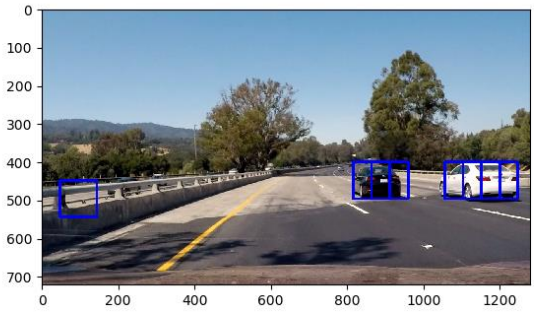
Sliding Window Search

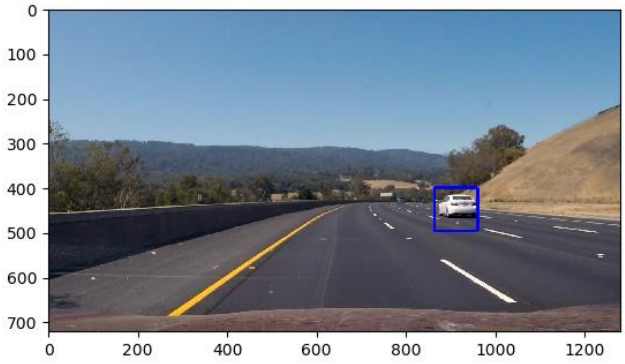
Now after we have trained our SVM classifier we need now to detect objects in the image so we can find the vehicle to do that we will apply a sliding windows with different scales after that we will apply to it

Below is images for applying one scale for the image

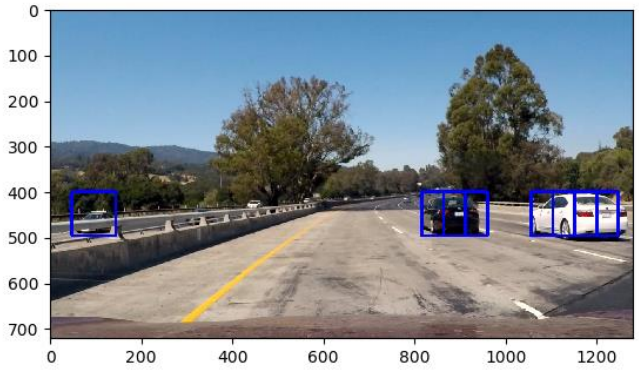




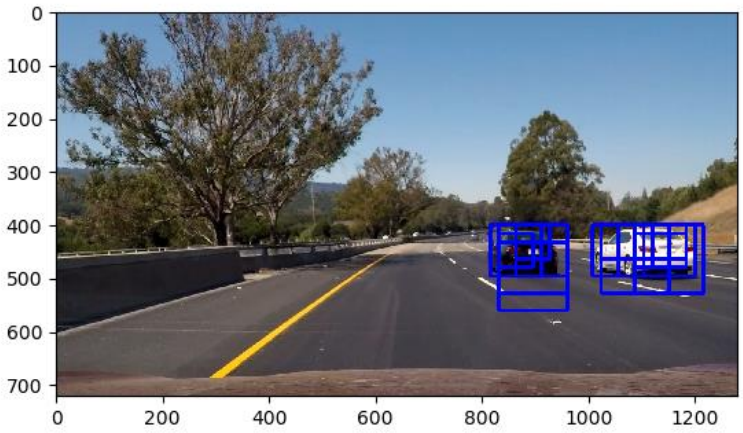


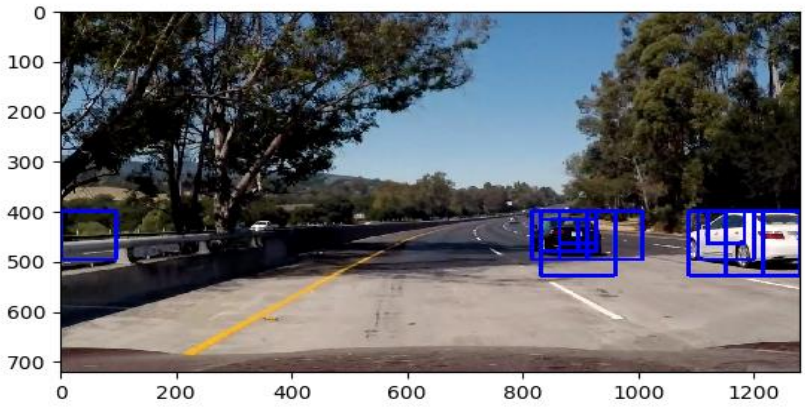

0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200

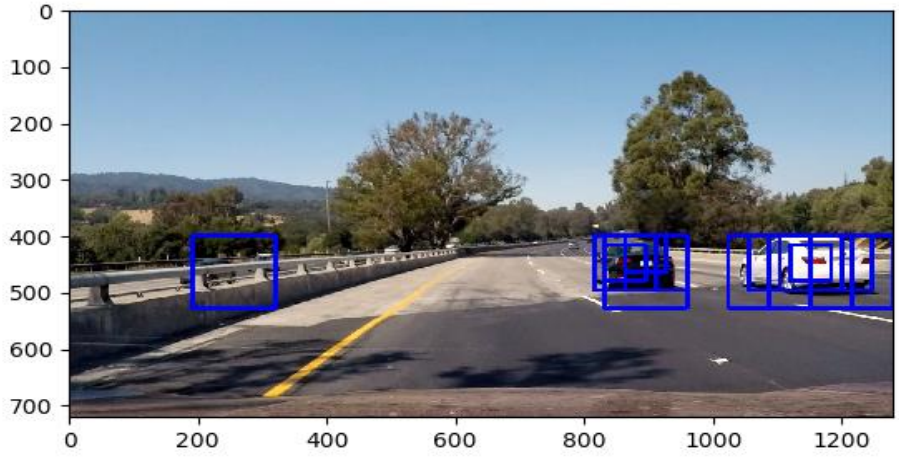

0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200


0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200

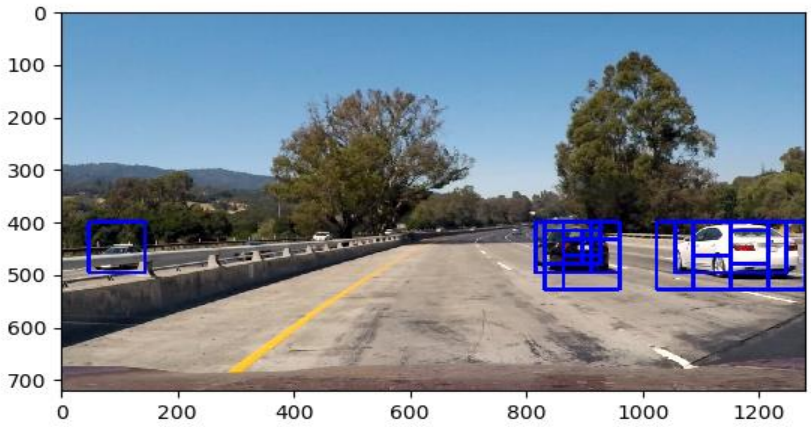
In the processing of the video we apply different scales so we can well cover the track
Below is images when we used different scales.






0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200


0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200

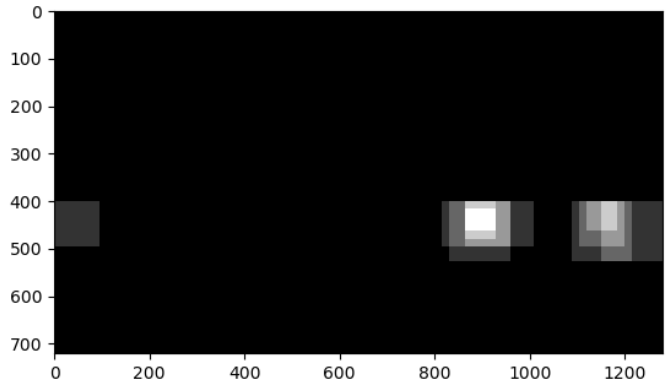


It seems to work well however sometime as the last image in a certain scale it detected a negative label as positive one to fix that we will apply heat map.

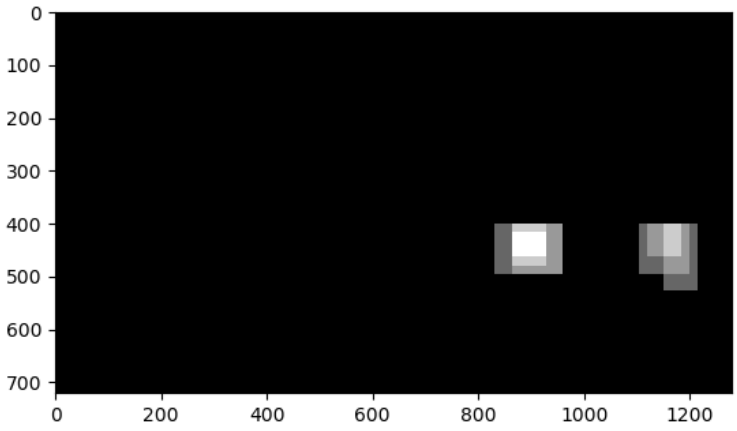
Heat Map

The idea behind heatmap is +=1 each time we detected an object while we move through different sliding windows with different scale. This way will make that if we have found an object once to be ignored and if found several time to be counted as one track. Below images to show different steps of the Heat map

The image below show that adding +=1 to each found track make some of them weak and other more strong



Now we will apply threshold to remove weak detections



Draw their equivalent on original image

DISCUSSION

Problems I faced while I am working:

Normalizing the data using `sklearn.preprocessing.StandardScaler` then transform it for me was not a good idea as after I did that I have found that the model capture vehicle and non-vehicle objects so I removed it seems work better now

This model has the following problems/Suggestions:

- It is very sensitive to the color so if we are night I don't think it will work fine
- We can perform better Algo to track it if we assumed that an object found in five frames consecutive expected to be found after that, as it is for example a car beside mine with some calculation I can expect where it will be next frame.
- Use deep learning instead of Classifiers

RUBRICS PONTS

All Rubric points are supposedly covered but to make sure I have not missed anything I will prove them here.

Explain how (and identify where in your code) you extracted HOG features from the training images. Explain how you settled on your final choice of HOG parameters.

Using some experimentation I found YUV is the best color spatial and I found the following params give the best performance

```
orient = 11
pix_per_cell = 16
cell_per_block = 2
check Training data section
```

Describe how (and identify where in your code) you trained a classifier using your selected HOG features (and color features if you used them).

Check function `PrepareAndTrainSVM`

`I have used the data trained as described above with SVM model`

Describe how (and identify where in your code) you implemented a sliding window search. How did you decide what scales to search and how much to overlap windows?

`find_cars_differentScales` `used to try different scales to find the object`

`find_cars_` `used to find vehicle for a specific scale`

Show some examples of test images to demonstrate how your pipeline is working. How did you optimize the performance of your classifier?

Done check Above

Provide a link to your final video output. Your pipeline should perform reasonably well on the entire project video (somewhat wobbly or unstable bounding boxes are ok as long as you are identifying the vehicles most of the time with minimal false positives.)

Check project_video_output and test_video_output videos

Describe how (and identify where in your code) you implemented some kind of filter for false positives and some method for combining overlapping bounding boxes.

Heatmap is used check section above

Briefly discuss any problems / issues you faced in your implementation of this project. Where will your pipeline likely fail? What could you do to make it more robust

Check Discussion section