Report Lorenzo Steccanella project P5 Vehicle Detection

The goals / steps made for accomplish this project are the following:

1. Dataset visualization
2. Feature extraction using the convert color, the Histogram of Oriented Gradients (HOG) , color histogram and concatenate the features.
3. Train and test an SVM classifier in order to detect vehicle images.
4. Implement a sliding-window technique and use your trained classifier to search for vehicles in images.
5. Implement a heat map for remove false positive
6. Estimate sequence of frame detection solution for smoothing and reduce false positive
7. Provide a Project_video output.

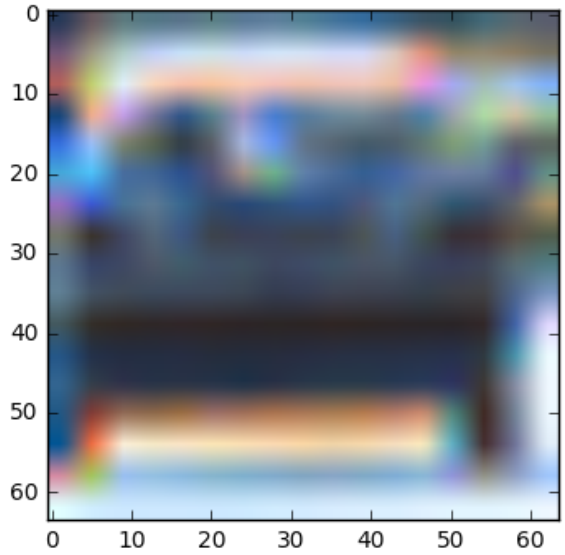
1 Dataset visualization:

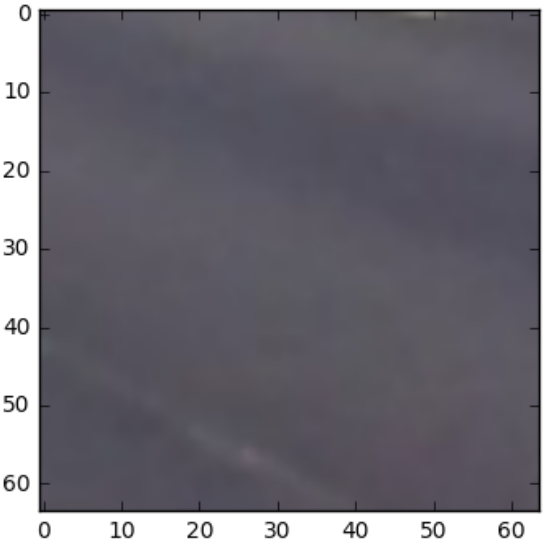
For this project the dataset Vehicle/ non-vehicle provided from the lecture has been used this dataset is composed by multiple images of car from different angle of view and from different non-vehicles classes.

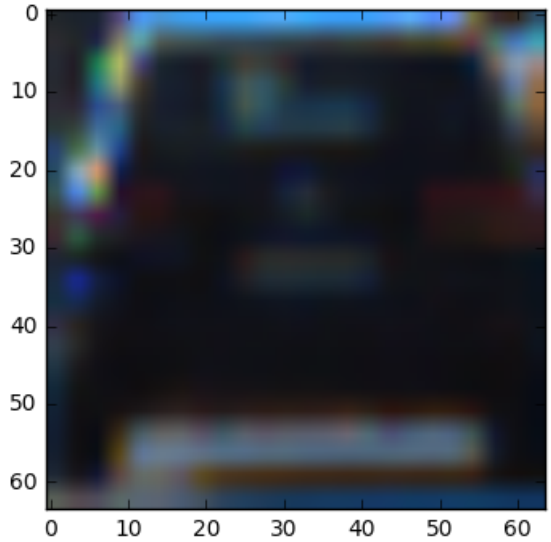
number of car images: 7883 size of each image= 64x64x3
number of non car images: 8967 size of each image= 64x64x3

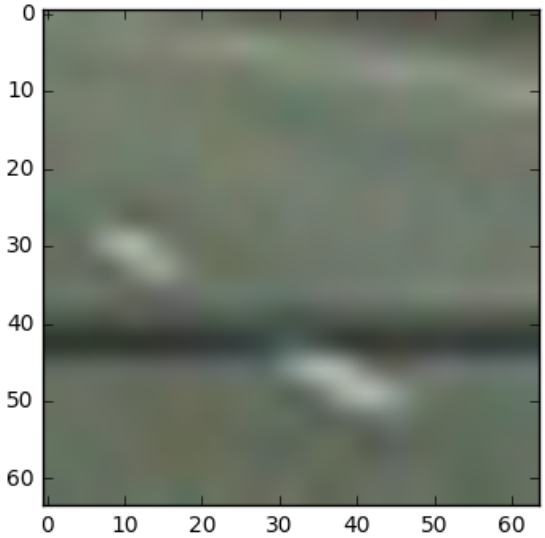
Example of 5 car and 5 non car images of the dataset.


0 10 20 30 40 50 60
0 10 20 30 40 50 60


0
10
20
30
40
50
60
0 10 20 30 40 50 60


0
10
20
30
40
50
60
0 10 20 30 40 50 60


0
10
20
30
40
50
60
0 10 20 30 40 50 60

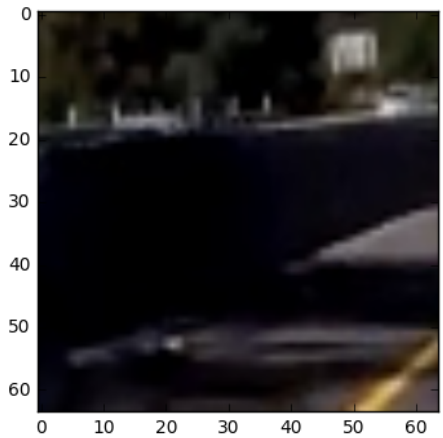

0
10
20
30
40
50
60
0 10 20 30 40 50 60

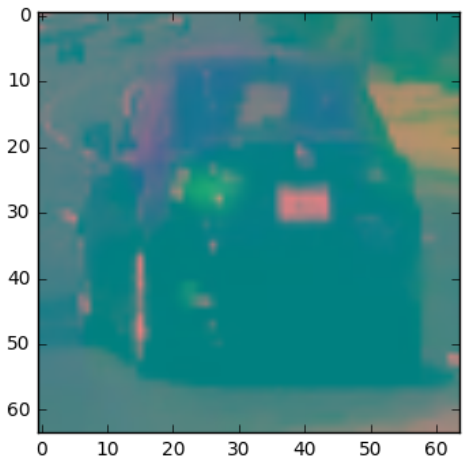
2 Feature extraction:

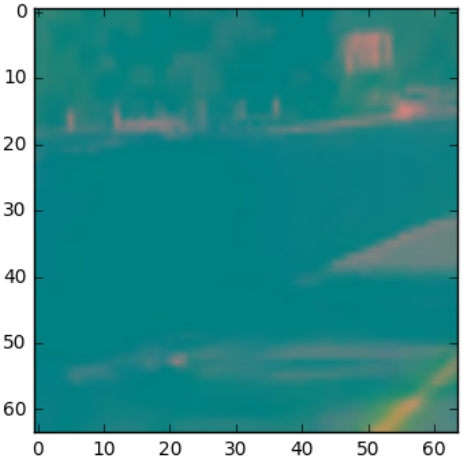
Visible at the 4 to 5 cells of the ipython rubric.

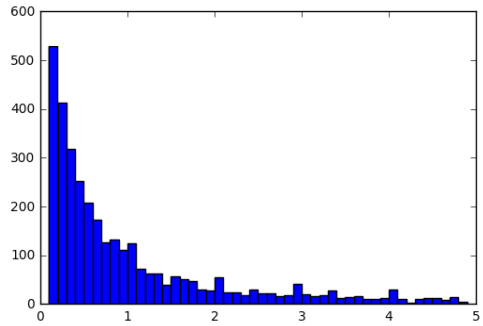
The feature selected for this projects are:

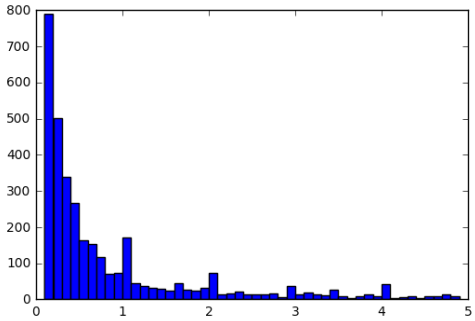
- First compute the convert color from RGB to YCrCb
- Extract the hog feature for each color channel (0,1,2) with a bins of 32, orientation = 9, pix_per_cell = 8, cell_per_block = 2
- bin_spatial the image to 32 x 32
- Apply a color_hist with nbins=32
- Concatenate the features extracted for hog, color hist, bin_spatial











3 Train and test an SVM classifier:

Visible at the 6th cell of the ipython rubric.

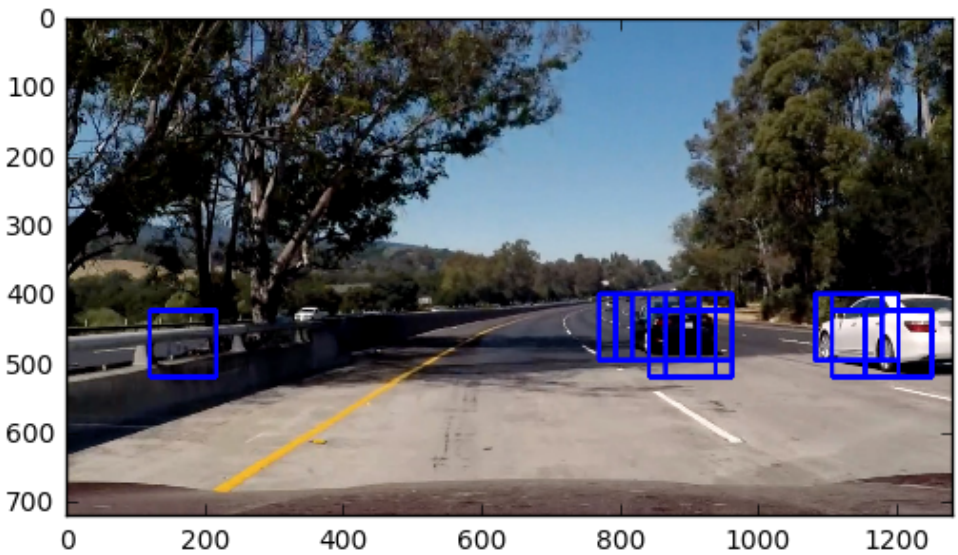
The extracted features have been normalized with a StandardScaler before being used to train the classifier.
For this project an SVM classifier has been implemented, since the classes are just two and spatially separated this classifier I think can perform at best. The dataset has been splitted for the 80% to Training set and 20% to Test set. The label has been created having 1 for indicating images with a car and 0 for images without a car. The performance of the classifier on the test set are: 0.9938.

4 Implement a sliding-window technique:

Visible at the 7th cell of the ipython rubric.

In order to locate the car in a image a moving window has been used. This moving window is not applayed to all the region of the image but just for the area from y=400 to y=656 that is the area realeted to the road. Each window box has been used with the svm to predict if it contains a car or not. This approach create multiple overlapping window that are usefull in the next step and the SVM create also some False positive.

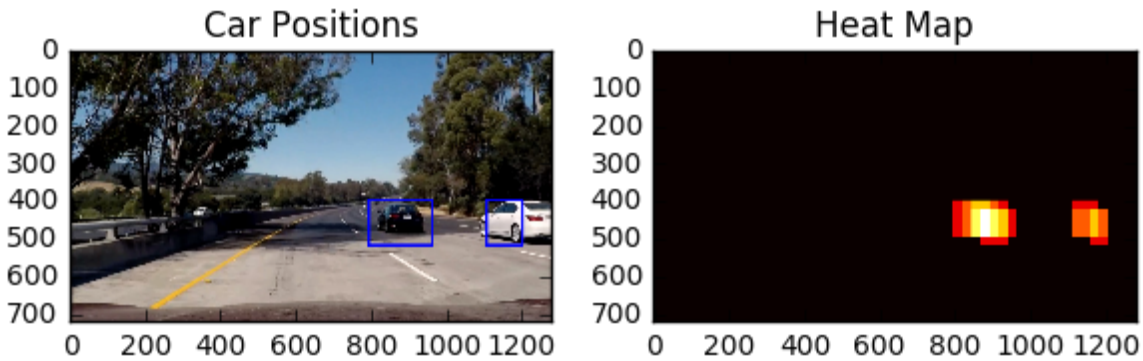
ystart	400
ystop	656
scale	1.5
window	64
cells_per_step	2



5 Implement a heat map:

Visible at the 8 to 10 cell of the ipython rubric.

In order to remove the false positive a heatmap has been created, that add 1 at each pyxel that has been predicted to have a car. Since the windows are overlapping we will have that the sum of 1 on the pyxel would lead to have higher values where the real car are. Thanks to this, with a simple treshold, that in this case I set to 1 we can remove the false positive.



6 Estimate sequence of frame detection solution for smoothing and reduce false positive:

Visible at the 11 to 12 cell of the ipython rubric.

The heat map in the video has been combined frame by frame in order to have a smoother detection and reduce the false positive.
The heatmap of 6 frames has been combined and a threshold of 5 has been applied.

7 Final video:

https://www.youtube.com/watch?v=FVzkxRjQYc0