Writeup

Vehicle Detection And Tracking Project

The goals / steps of this project are the following:

- ·0 Perform a Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set of images and train a classifier Linear SVM classifier
- ·1 Optionally, you can also apply a color transform and append binned color features, as well as histograms of color, to your HOG feature vector.
- ·2 Note: for those first two steps don't forget to normalize your features and randomize a selection for training and testing.
- ·3 Implement a sliding-window technique and use your trained classifier to search for vehicles in images.
- ·4 Run your pipeline on a video stream (start with the test_video.mp4 and later implement on full project_video.mp4) and create a heat map of recurring detections frame by frame to reject outliers and follow detected vehicles.
- ·5 Estimate a bounding box for vehicles detected.

Here I will consider the rubric points individually and describe how I addressed each point in my implementation.

Writeup / README

·6 **Provide a Write-up / README that includes all the rubric points and how you addressed each one. You can submit your writeup as markdown or pdf.**

Yes, I am submitted Write-up that includes all rubric point in pdf format.

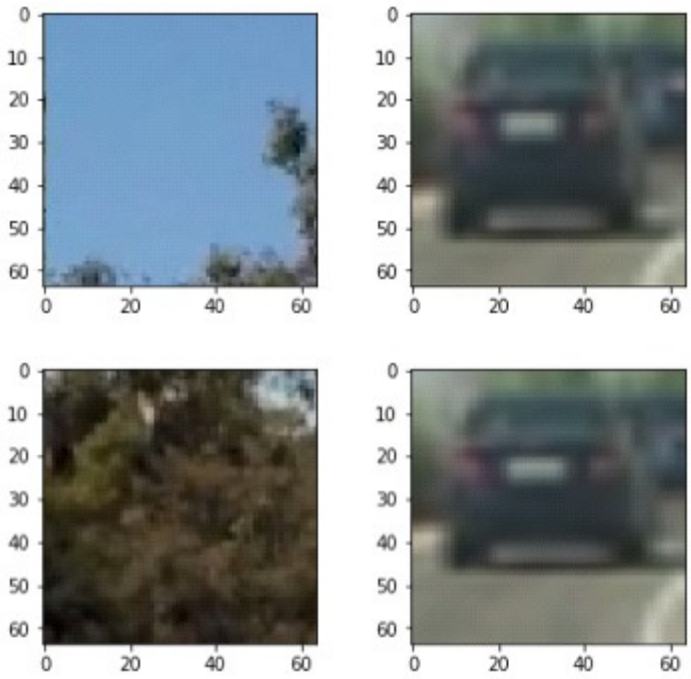
Histogram of Oriented Gradients (HOG)

·7 **Explain how (and identify where in your code) you extracted HOG features from the training images.**

I used “**from skimage.feature import hog**” library to extract feature from the training image which accept following arguments:

Image: I have taken car and non car Images from the data provided and performed the hog transformation on it and the output for that is as below:

car images noncar images



Orientations, **pixels_per_cell**, **cells_per_block:** I have to try different combinations which provided me with the best results

Block Normalization: L2-Hys is used which is L2-norm followed by clipping (limiting the maximum values of v to 0.2) and renormalizing

Here is an example using the following parameters i have calculated the HOG visualization:

color_space = 'YCrCb'

orient = 9 # HOG orientations

pix_per_cell = 16 # HOG pixels per cell

cell_per_block = 2 # HOG cells per

hog_channel = 'ALL'

spatial_size = (16,16)

hist_bins = 32

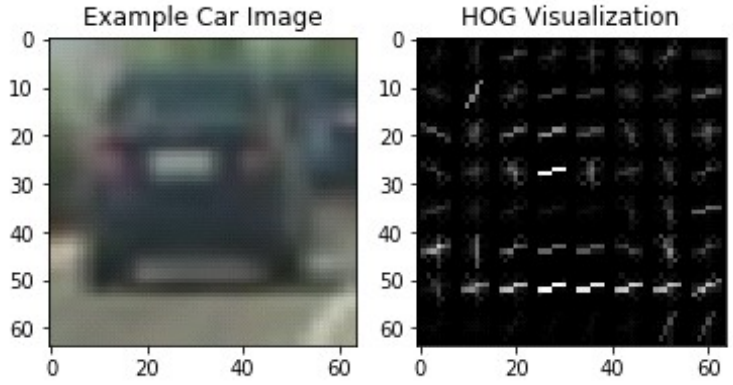
spatial_feat = True

hist_feat = True

hog_feat = True

y_start_stop = [None, None]

And the output image after calculating the HOG visualization is as below:



·8 Explain how you settled on your final choice of HOG parameters.

I tried various combinations of parameters.

Innitially i tried with 'RGB', 'HSV', 'LUV', 'HLS' combinations for which gave me accuracy values of 97.5%, 97.8%, 96.8%, 98.0% respectively. But was unable to get the satifactory output using the these then finally i tried with YrCbCr color space.

I found out that 'YrCbCr' is giving me accuracy of 98.65%. So I have to keep feature size below 1836 (as suggested in the classroom that ideal feature size for SVM is 20% of total training images). Increasing the Orientations is increasing feature size but at same time increasing the accuracy. To decrease the feature size, I increased the pixel per cell from 11 to 9 in my final outcome. I used the spatial feature size of (16,16). And using all these trials and error i was able to achieve an accuracy of 98.65%.

·9 Describe how (and identify where in your code) you trained a classifier using your

selected HOG features (and color features if you used them).

I have used LinearSVC() to train the:

- 10 Hog Features
- 11 Color Features
- 12 Spatial Features

Following parameter are used:

- 13 color_space = 'YCrCb'
- 14 orient = 9 # HOG orientations
- 15 pix_per_cell = 16 # HOG pixels per cell
- 16 cell_per_block = 2 # HOG cells per
- 17 hog_channel = 'ALL'
- 18 spatial_size = (16,16)
- 19 hist_bins = 32
- 20 spatial_feat = True
- 21 hist_feat = True
- 22 hog_feat = True

I used Standard Scalar to Normalize the features across images. Then Images were divided into Training and Test dataset in a rato of 8:2. I got the below accuracy using a linear kernel:

Using: 9 orientations 16 pixels per cell and 2 cells per block

Feature vector length: 1836

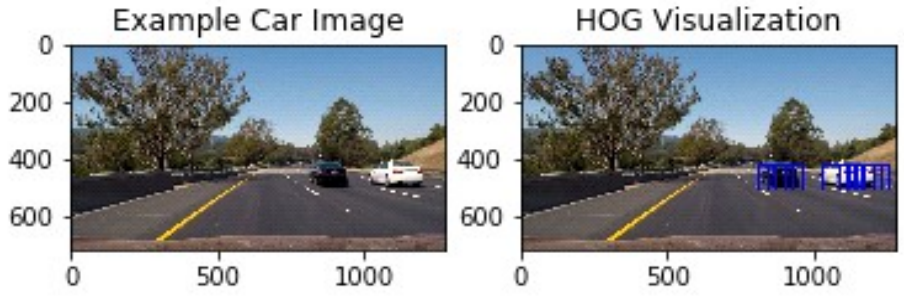
8.36 Seconds to train SVC...

Test Accuracy of SVC = 0.9865

- 23 **Show some examples of test images to demonstrate how your pipeline is working. What did you do to optimize the performance of your classifier?**

The following is the output of the pipeline where i am getting the different window overlaps

which i have applied in the slide _window() function for which I am further taking the heat map of the overlaping points .



My Performance depends on:

·24 Limiting the feature size to about 20% of the training data.

·25 Heat maps are useful for removing Threshold.

.

Video Implementation

·26 **Provide a link to your final video output. Your pipeline should perform reasonably well on the entire project video**

I am providing the link to my project_output as below:

https://github.com/vikasbij/CarND-Vehicle-Detection/tree/master/videos

·27 **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.**

·28 Firstly I extracted all the false detection frames from the project video and test video.

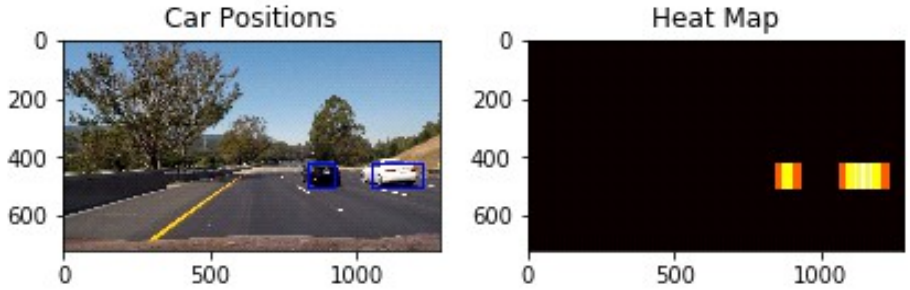
·29 Then recorded the positions of positive detections in each frame of the video. From the positive detections I created a heat map and then threshold that map to identify

vehicle positions.

·30 Then I used **scipy.ndimage.measurements.label()** to identify individual blobs in the heat map.

·31 Then I assumed each blob corresponded to a vehicle. I constructed bounding boxes to cover the area of each blob detected.

Finally i got the heat map for the images as below:



Discussion

·32 The pipeline I have used in this project tends to do poorly when areas of the image where the color lighting is brighter than usual part of the image.The pipeline usually classifies the white cars pixels as false-positives. This issue could be resolved by adding more images to the vehicle dataset.

·33 I would also like to try implementing deep-learning for vehicle recognition in future. Or may be using YOLO can be an area of intrest for me for implementing this.