CarND Vehicle Detection Project

By: Jeffrey Lutz

The goals / steps of this project are the following:

- Perform a Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set o f images and train a classifier Linear SVM classifier
- Optionally, apply a color transform and append binned color features, as well as histograms of color, to the HOG feature vector.
- Normalize the features and randomize a selection for training and testing.
- Implement a sliding-window technique and use the trained classifier to search for vehicles in images.
- Run the pipeline on a video stream and create a heat map of recurring detections frame by frame to reject outliers and follow detected vehicles.
- Estimate a bounding box for vehicles detected.

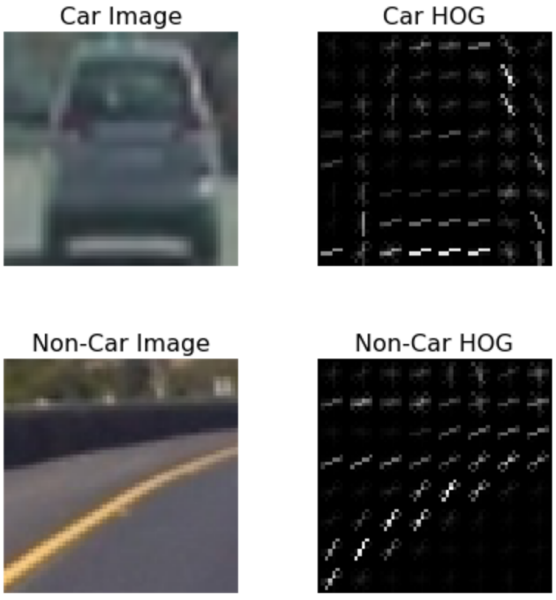
Rubric: Write-up (PDF form)

This document fulfills the requirement for a project write-up

Histogram of Oriented Gradients (HOG)

In code section #4 of the notebook file CarND-Vehicle-Detection.ipynb, I created a function called: get_hog_features(). This function extracts the features and optionally creates an image. Code block 5 exercises and displays the corresponding image for both a car image and a non-car image.

Example of car and non-car image with the corresponding HOG image



In code block #7 of the notebook, I explored all of the available color-spaces, orientations and block sizes. I considered the extraction time duration, train time

and accuracy as the criteria for goal of most accurate with the least computational cost. In the notebook I documented this accuracy vs. performance tradeoff. Configuration label #17 has the Colorspace: YUV, Orientation: 11, Pixels/Cell: 16, Cells/Block: 2 and HOG channel: ALL appears to have an accuracy of 99+% with a small training time of 1.1 seconds. The extract time of 53.81 seconds is more nearly double the time duraction of the minimum extract time of 28 but the accuracy and training time are more important to the value of the trained network.

Rubric: Classifier Training

In code block #8, a linear SVM classifier is trained with the HOG parameters that appear to be optimal from earlier experimentation.

Rubric: Sliding Window Search

The function find_cars() defines the sliding window search logic of the notebook. This function is located in code block #9 in the notebook. The method used for stepping through the windows of the image searching for cars was to compute windows with a size of 16 pixels per window and a y start location of the horizon value of 400 and going to the near bottom of the image using y stop of 656 pixels. I scanned from the left edge to the right edge of the image for x value consideration.

The overlap of the windows was 50% the size of the window. The computational cost wasn't too bad with the benefit of not missing car identification.

Here is an example of a car window image:

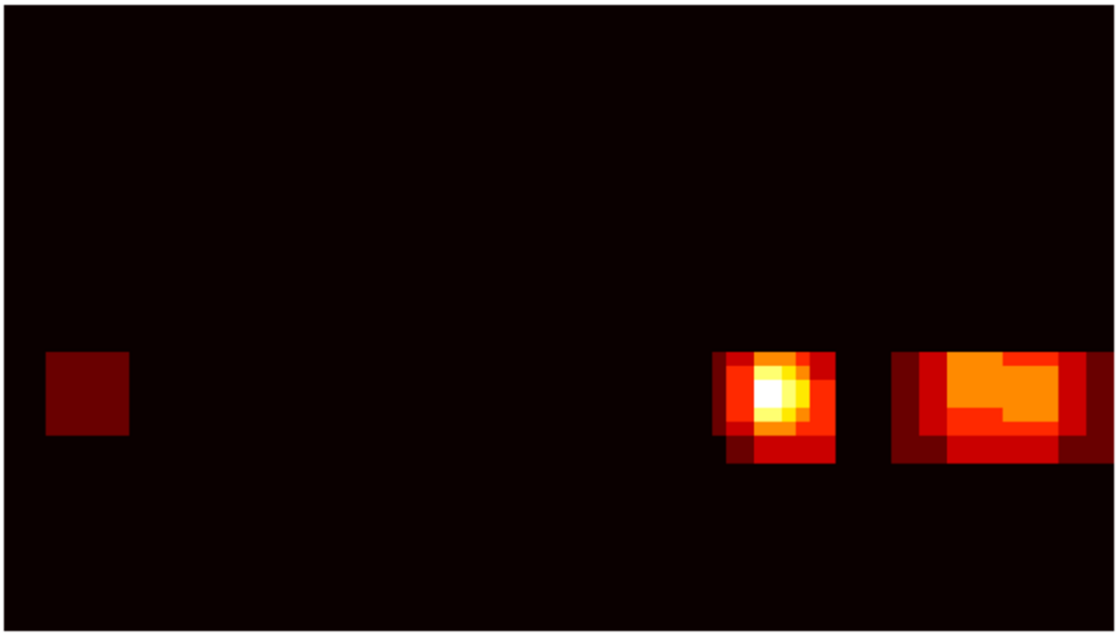
Rubric: Video Implementation

The following video (project_video_out.mp4).shows the performance of the detection system. The code for removing false positive vehicle identification was captured in code block 25 of the notebook. The actual reduction of false positives was implemented by using the scale values listed in the code block.

In code block 18 and 19 of the notebook, I explored the heatmap of the vehicle detection. The heatmap helped to combine detection boxes and clearly show the general location of each vehicle. Below is a picture of one frame.

Image frame with bounding boxes

Same image frame with heatmap



Then in code block 22, the image had threshold applied and code block 23 shows the same car detection frame but with gray boxes.

```
2 cars found
```

Lastly, the final bounding box for each vehicle is seen below for the same image frame within the code block 24's function "draw_labeled_bboxes()

```
(-0.5, 1279.5, 719.5, -0.5)
```

The final image pipeline is contained in code block 25. Essentially, the eight vertical rows are scanned for rectangles and then combined. Then, the heatmap processed

for reduction of bounding boxes for each vehicle down to a single bounding box. Then, the final bounding boxes are drawn on the image frame.

Summary

The vehicle detection project has a number of unique challenges. First was to develop an understanding of the histogram of gradient (HOG). Then, to explore and select the optimal settings for the HOG function. It was really fascinating to see how the HOG function distilled down the image into a new representation that contained only the essential components of the image.

The next concern was to work through the find car function to identify the optimal sliding window configuration. Focusing on only the ground (ignoring the sky) was an easy determination. However, choosing the window sizes at various locations vertically was more challenging. Using trial and error was most useful. I was surprised at how small the number of steps in the vertical direction of the windows. This project really increased my understanding of how to train and use classification detectors such as the linear SVM classifier.