Vehicle Detection

by K.Vikraman

Goals of this project :

1. Perform a Histogram of Oriented Gradients (HOG) feature extraction on a labeled training set of images and train a classifier Linear SVM classifier
2. Additionally apply a color transform and append binned color features, as well as histograms of color, to HOG feature vector.
3. Implement sliding window technique and implement SVM classification in each window to find cars.
4. Add heat maps to remove false positive

Step 1: Data preparation and feature extraction:
Data preparation steps:

1) Convert into suitable output range (0-1).
2) Accumulate car images and non-car images separately
3) Randomly shuffle the data
4) Split the data

The first step is to choose what features to include and what features of an image to be excluded. Different colorspaces are useful in a different ways. After investigating different colorspaces, I got maximum accuracy in 'YCrCb' colorspace.

After converting the image to YCrCb colorspace, I analysed HOG features of each channel of an image and found that HOG features from all channels useful.

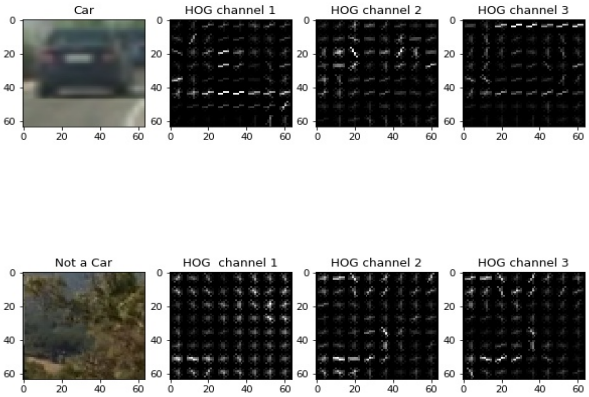

Figure 1 : HOG Features

Color histogram features:

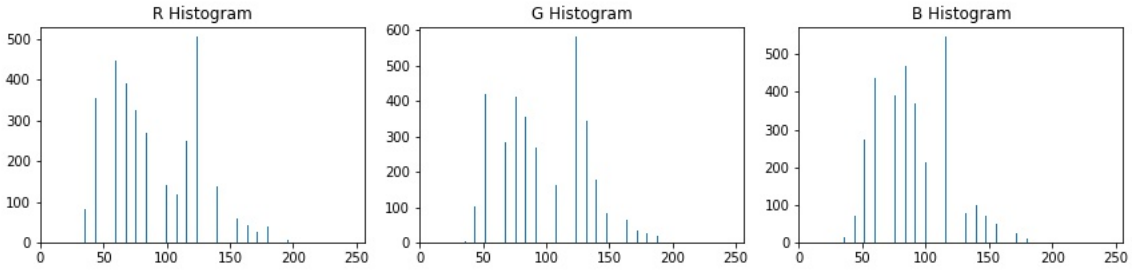

Figure 2 : Color Histogram for each channel

The above are the histogram values of the given image. Features of colorspace gives us important information about cars with different colors. To investigate further I analysed information from other colorspaces. The following is the plot of histogram features of the same image from YCrCb colorspace.

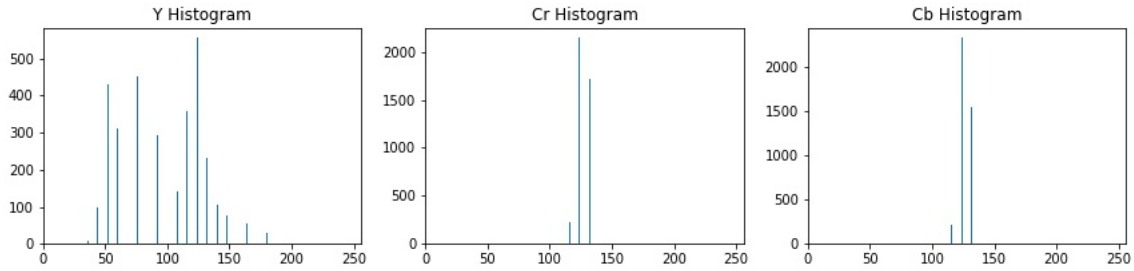

Figure 3 : Histogram for YCrCb colorspace

There is a stark contrast and YCrCb colorspace gives maximum accuracy. Hence, I choose YCrCb colorspace.

Step 2: Training a classifier :

Linear SVM is a powerful and quick classifier, much suited for our purpose.
I got an accuracy of 99.3% in the test set.

Step 3: Search windows :

Applying search window technique essentially finds whether a given patch has a car.

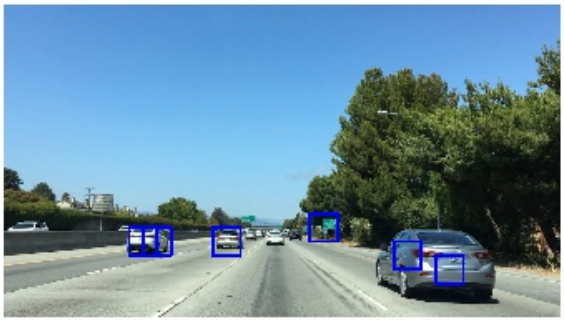

Figure 4 : Implementing search window technique

But it fails to identify car as whole and does not remove false positives.

Step 4: HOG subsampling :
The images in the training set are of shape 64x64 pixels. However the images used in our pipleine are of shape 720x1280 which is much larger. Calculating the HOG features will reduce the efficincy and will not account the features in the neighboring pixels. In order to improve the efficiency, I first calculate the HOG features for the overall image. Following that, I extract the HOG features for that particular sample alone.

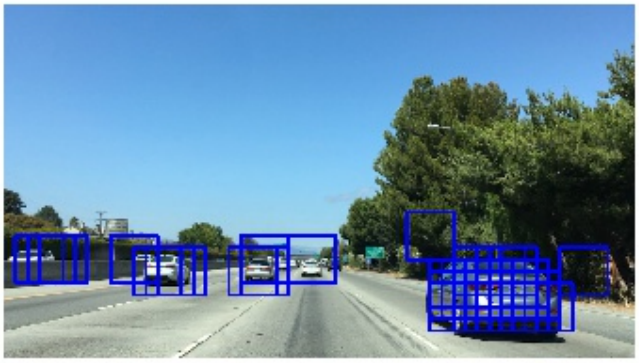

Figure 5 : HOG Subsampling

This increases the accuracy. The next step is to club the group of windows and remove false positives.

Step 5: Adding Heat Map :

Adding heat map with a threshold drastically removes the false positives and groups all the windows into a single window identifying the car.

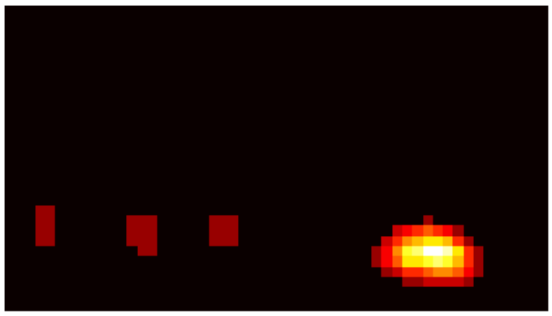

Figure 6 : Heat Map

Figure 7 : Vehicle detection

Since farther cars appear smaller and nearby cars appear larger, I used different scales to search an image and appended all the features, which improved detection.

Discussion:

Though this method works fine in the given video set, it has few drawbacks.

1) Multiscale approach improves the efficiency at the cost of increased computational time. One iteration approximately takes 3 secs in a Core i5 CPU, which is too long to beneficial for realtime processing.
2) The training set is inadequate to accommodate all possible real life situations. For instance, detection in a tunnels, snowy conditions where some cars are likely to be covered with snow. In these cases the algorithm might classify the car as True Negative.
3) Cars in the other side of the road are seldom detected.