# Project 5 Vehicle Detection

**Histogram of Oriented Gradients (HOG)**

Read the images from the Car and noncar folders. All the images around 5000 for each car and non car are used for the training the classifier.

I have used YcrCb color space and HOG parameters of orientations =8, pixel per cell = (8,8) and cells per block = (2,2)

2. Tried carious combination of HOG parmeters like orientation of 11 and 4 pixel per cell. But It gave less accuracy  when compared with the selected parameters.

3. Exttacted hog features have been used to train the classifier. Before training data is normalised using StandardScaler().

Linear SVC is used to classify the data. Classifier give acuuracy of 98 percent.

Trained data are splitted and shuffled before extracting the features.

Please look at the Cell 6. Below is the parameters used.

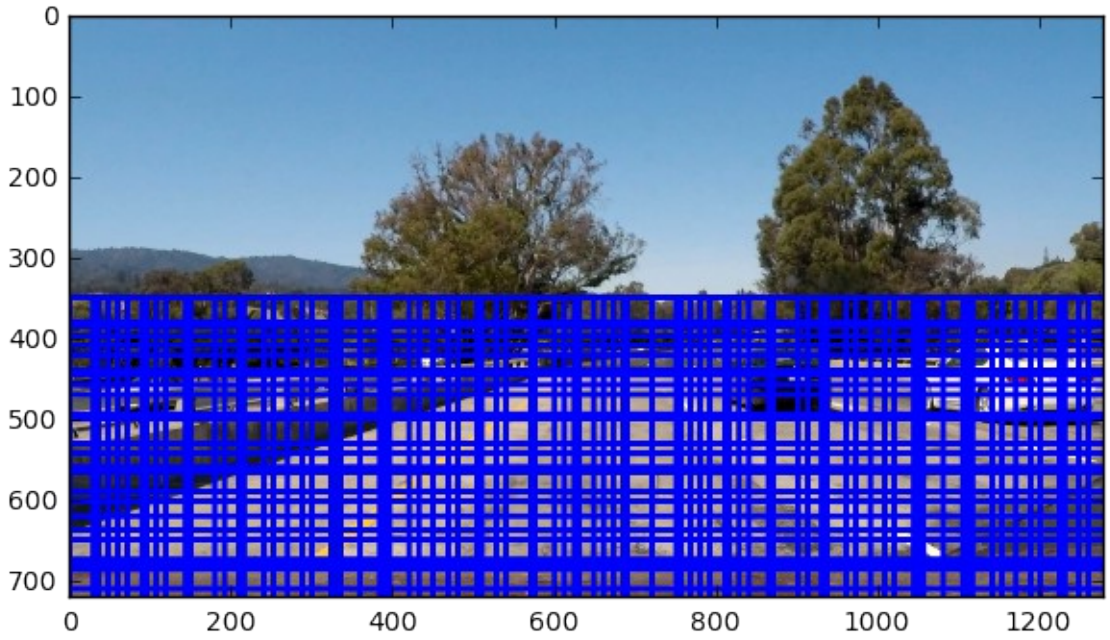
```
color_space = 'YCrCb' # Can be RGB, HSV, L
orient = 8  # HOG orientations
pix_per_cell = 8 # HOG pixels per cell
cell_per_block = 2 # HOG cells per block
hog_channel = "ALL" # Can be 0, 1, 2, or "
spatial_size = (16, 16) # Spatial binning
hist_bins = 32    # Number of histogram bi
spatial_feat = True # Spatial features on
hist_feat = True # Histogram features on o
hog_feat = True # HOG features on or off
#y_start_stop = [None, None] # Min and max
y_start_stop = [350, 710]
#def extract_features(imgs, color_space='R
```

Initially I tried with RGB color space too. But it worked well when training sample was around 500. It recognized car features, but skips frames frquently. As detection was not so smooth and false positives are also more with RGB, I tried on HSV and YcrCb color space. Both are working well, but YcrCb gave better results.

**Sliding Window search:**

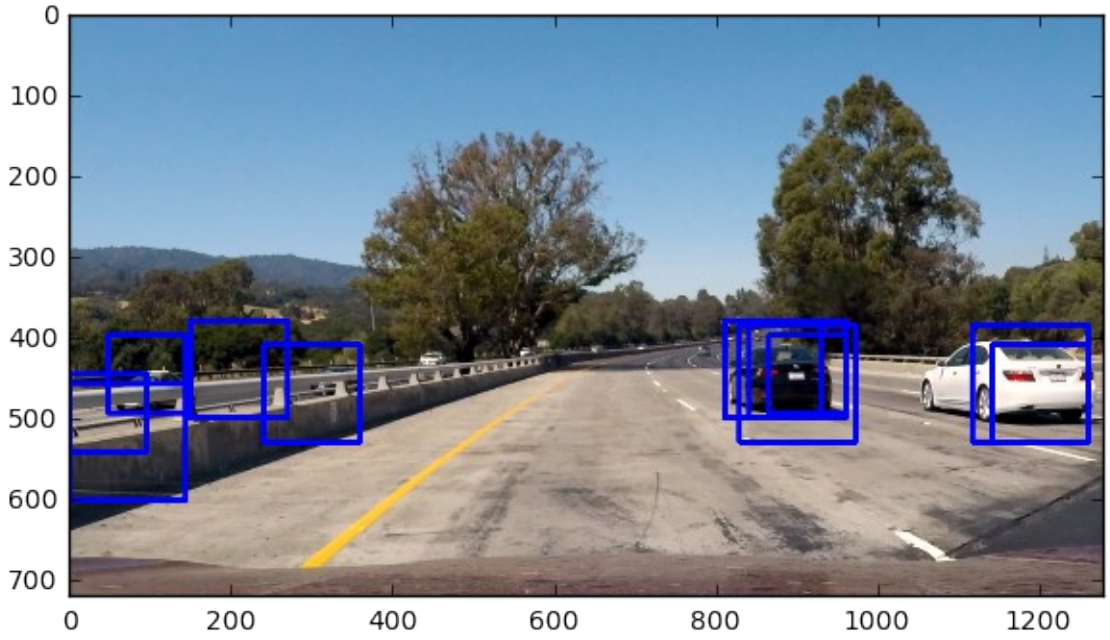
Searched for the features in the lower half of the image only.(Y 350-710). Tried different sizes and the overlap of the windows. When tried with 48, 48 size window, detected cars well but the extracted features are too local and cold not be meged with larger window size(at least fot the my code., as I have not averaged the windows for smoother and global car capture). Lasty I have used larger size windows like 96, 120, 192,144  and captured the car well. Reason may be training data contains more images of the whole car-body.

Below is the image overlapped with all sized windows.

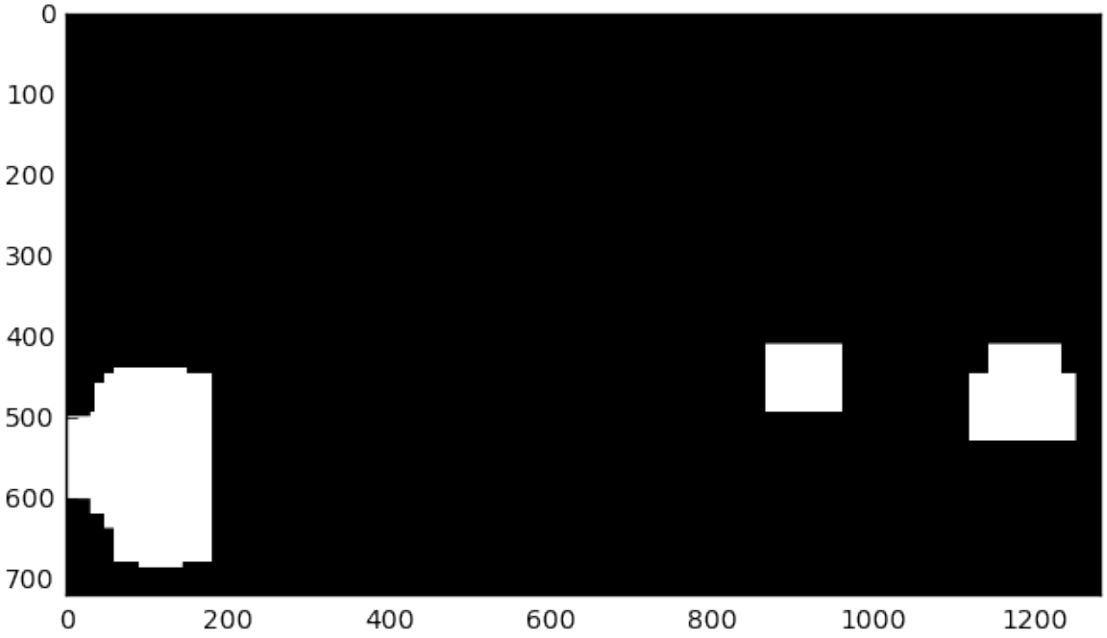



4. I searched on two scales using YCrCb 3-channel HOG features plus spatially binned color and histograms of color in the feature vector, which provided a nice result. Here is an example image:

   I mage has a false positive too. But it gets removed when heatmaped and thresholded on the few frames. Below image is before applying threshold.



5. Once window serch get results for windows where the fetures matched, heatmap image is created. Please look at Cell 19 and 20


Above is the image for heat map. Above is the output for sigle frame. And if fetaures extracted at the same location it adds to the heatmap.

I have added the heatmaps for 5 consecutive frames. Threshold of 3 is used to remove false positives(as in the left side of the image).

Boxes are averaged for the one frame using draw_labeled_bboxes() as shown in the Cell 9. Windows are averaged for single frame and not over the frames. Output boxes are overlayed on the given frame to get the result. Below is the output for the sigle frame.

Hopefully false positive get removed after few consequtive frames.


**Video:**
Code is implemented on the video in the Cell 17.
Video image are divided by 255 before extracting features as the trained classifier also takes the image in this format. Output looks not optimum. Windows are not smooth and vibrating much. But still code detects the car fairly well.

**Discussion:**

It was tricky to decide the size of the window. Initially when tried smaller sized windows of 48*48 and 96*96 and worked on the other varibles. But it gave poor results. When I shifted to larger windows (96, 120, 14, 192), code started behaving well.
Other variable was heatmap threshold which is also a deciding factor. When I was added heatmap for just two frames and threshold of 2 has not helped much. Then I tried with 3 or more frames with threshold of 3, which gave better solution and detedted the cars well and removed the false positives.
Code results shows the vibrating windows. Smooth movement of the window can happen by averaging the windows over the frames and displacing it smoothly without jerks if windows are clser(ie. of the same car)
Code is not robust enough to seperate the two more cars moving closely. There may be some methods to seperate the cars and track them seperately.Present code gives vibrating windows in such closed cars scenarios.

Also code detect many false positives in shadow conditions and hence likely to fail in night or poor weather conditions. Workaround for that may be adding more data in different enviornment conditions.
I thing making the code to detect the noncars and cars too will help. Later we can have robust code to differntiate it if overlapped.