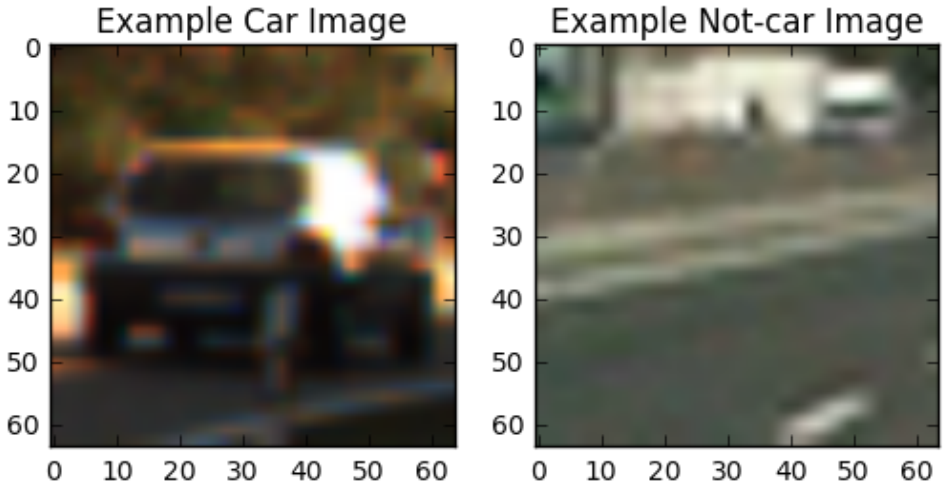
Vehicle Detection Project 5

The goal of this project is to detect the vehicles which are in front of the car, so that it can used for self driving.
Vehicle Detection done in this project involves following steps:

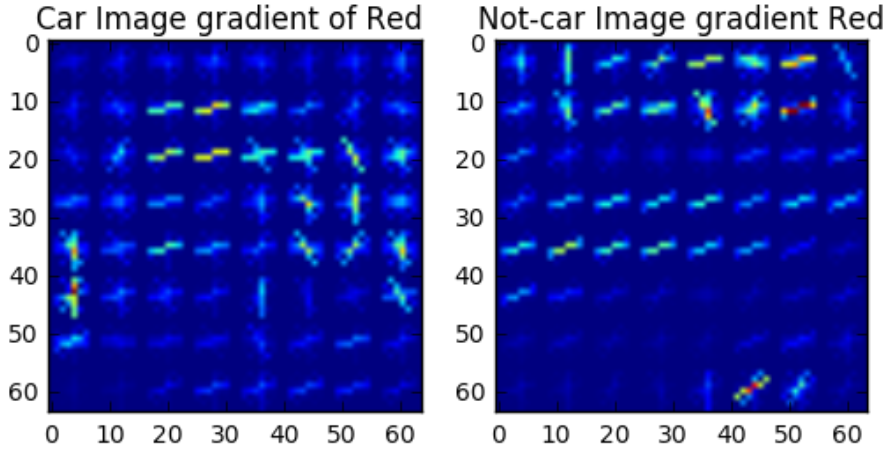
- Feature Extraction for training and testing
- Machine learning training and testing
- Sliding window approach to detect the presence of car
- Heat map approach to pin down exact position of car
- Applying pipeline software on video

Feature Extraction for training and testing:

Two classes of images are needs for this project. One set of examples belonging to the view of cars from their back or side to back views and second set of examples belonging to the objects or shapes which are not cars but are visible on road. Example of each class are shown below:



Features are extracted for each image belonging to both class. A function called extract_features has been used. Gradient features are as shown below as image.A histogram of these gradients is one of the features.



The get_hog_features() function is used to extract the Hog Features. This function has two parts of implementation one is extracting HOG features alone as feature vector and the other is extracting HOG features as well as obtaining the Gradient image which was used to obtain the HOG. Skimage features library has hog() function which takes following parameter as input:

- ➔ Image channel on which the HOG needs to be extracted. One of the three channels of HLS are being sent by the calling function one at a time
- ➔ Parameter Orient- Number of bins used in the histogram
- ➔ pixels per cell-number of pixel in each cell being considered for computing gradient
- ➔ Cells per block- Number of cells in each block on which the final gradient of the block is finalized

Orientation bins being considered in this project are 9 pixels per cell are8 and cells per block are 2. Depending on the number of channels used the get_hog_features need to be called once or thrice and all the features collected at the calling function and made as a single vector using ravel of numpy

The other features used are color histogram and spatial binning which are explained in the class video and the color space used was YUV with all the three channels being used for HOG.

Machine learning training and testing:

Linear Support Vector Machine Classification was used for classification and 20 percent of samples from the data was used for testing while the remaining was used for training.

Sliding window approach to detect the presence of car:

Sliding window approach with multi scale overlapping window was used for scanning the test images for potential areas containing car. The result of multiscale overlapping window search is attached below

Improvement of reliability: At this juncture of testing different options of features were tried out with only hog as features, hog+color histogram as features, hog+ spatial binning , color histogram+spatial binning and hog+color histogram +spatial binning. HOG+Color Historgram+Spatial binning was found to have more accurate findings than others.

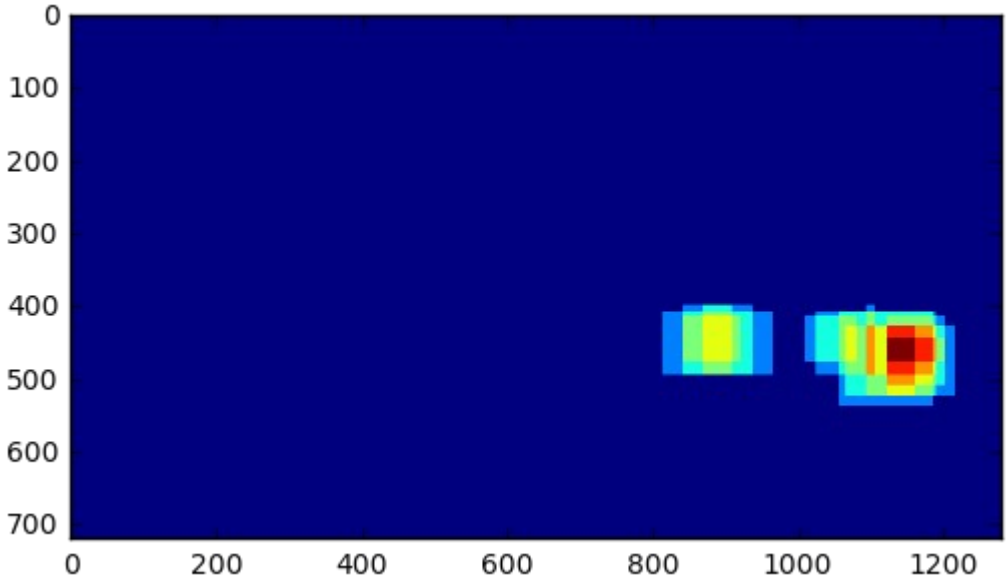
Color space of RBG, HSV ,HLS and YUVwere tried out and RGB tends to have more false positives than HSV and HLS while HLS seems to have better performance than HSV though this claim is debatable as there is very narrow improvement felt by using YUV. So YUV color space was used.

Regarding the bin size of Color Histograms 16 and 32 bin sizes were tried out and 32 size seems to

have better accuracy on testing. For Spatial binning spatial window size of 16x16 and 32x32 were used and the latter seemed to have improvement on test accuracy.
So Finally the HOG+Color Historgram+Spatial binning on YUV color space with 32 bins for color Histogram and 32x32 window sized spatial binning not only reduced false positives but also did the detection of vehicles in a reasonable way.

Heat map approach to pin down exact position of car

To remove false positives and to remove multiple windows on the same car heat map approach , as described and explained in the course work, has been used. Thresholding on the heatmap value was used to obtain the potential windows containing cars.
Here is an example of Heat map image :



Applying pipeline software on video

The afore mentioned techniques with running average of heat maps to fit in best window for cars was used as pipeline software on the video which is stored as project_video_output

Discussion

Problems Faced:

- Fine tuning of the parameters of scaling window were done on limited number of test images 6 to be precise but had to be varied slightly on the project video to improve accuracy.
- Some False positives can still be seen in the output video which is hard to eliminate with the available number of steps implemented in the pipeline
- In some areas the white car is hard to detect by the pipeline as the scanning windows scan the car as different parts

Likely failure of pipeline:

- There is a high chance of failure in scenarios where there is no divider between two lanes since the model is mostly trained by side view and back view of car
- More examples are needed to generalize the SVC model obtained to avoid failure of detection for example this model cannot detect tracks and other kind of vehicles
- Even applying scanning for a limited area of image takes longer time which can be a problem when the pipeline is used for real time implementation.

Potential Improvements:

- reducing the feature size by removing unwanted or less likely parts of the features by using decision tree or other dimensionality reducing algorithms like Principal Component Analysis(PCA)