Vehicle Detection and Tracking

HOG Extraction method and Parameter Choosing

1. I used the hog method from *skimage.feature* to get hog features.
2. I used the orient of 12, because 12-orientation is enough for car shape, more than this will make the feature too large.
3. I used the pix_per_cell of 8, that is 1/8 of 64. I want the classifier to detect the unique shape of the car such as windows, side mirrors, and wheels.

Training Classifier with selected HOG features

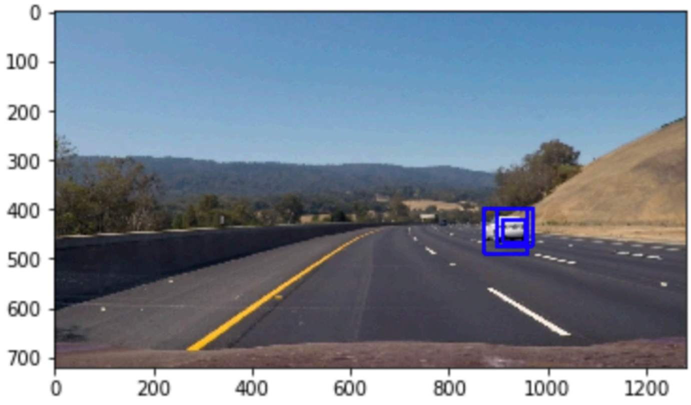
1. The amount of cars date set is more than that of notcars, so we need to downsample the cars data set to make it even. The final amount of each data set is 8792.
2. I use spatial feature and hog feature. I choose not to use color feature because I worried that gray cars in training set will lead the classifier to mistake the road as cars.
3. I used linear SVC here, since tutor from Udacity video said its already good.
4. GridSearchCV was implemented to improve the SVC performance.
5. All features are transformed by *StandarScaler* to be 0 mean and standard deviation. The scaler was trained solely on training data to avoid a peek into answer.

Finding images through Sliding Window Search

1. Windows are only located between y:400~656, x:600~None. The limits of y can help us avoid trees, skys, mountains.
2. Sliding windows are based on 6 sizes including (45,45),(75,75),(80,80),(90,90),(120,120),(180,100), so that I can detect cars far and near.
3. Different size windows were limited to a certain range of xy space. Window <=50 was limited to y:400~470. 50<Window<=85 was limited to y:400~550. The others are limited to y:400~656.
4. Overlap is (0.5,0.5). This parameter is adjusted by experiment.

Test images and Performance Optimization

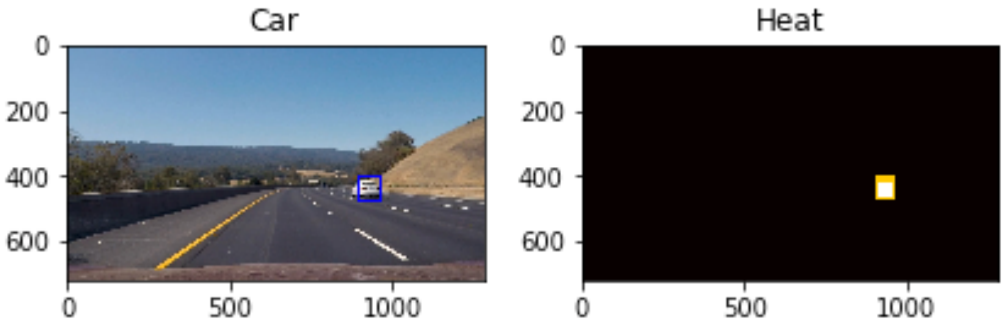
1. Test images:



2. I mainly optimize by adjusting features, window sizes and overlaps to get here. In the beginning, the classifier was having trouble in recognizing white cars when I was using HSV, then I changed to YCrCb, things get better. After that I mainly have trouble in detecting the fifth image where only a white car is in the far right lane position. A change in window size help this out. The most difficult problem is negative positives that keep showing up all the time, I improve it by adjusting overlaps and window size. The pipeline seems to do a good job in test images, but there are some negative positives in video I can’t erase.

Heat map filter

1. I used *label* method to filter out negative positives. Compared with test images above, we know that the label method is useful.



Problems and Potential Improvements

1. Different window sizes could be implemented in different section to reduce computation.
2. HOG could be implemented only once to reduce computation
3. Although we combined different features together, but their lengths are different, feature with longer length may have a bigger impact on the classifier.
4. Even though we improve the accuracy to 99%, the test image is large that with too many windows. The probability that there is no single negative positives through the whole video is small unless we can improve it into 99.99%.