Vehicle Detection

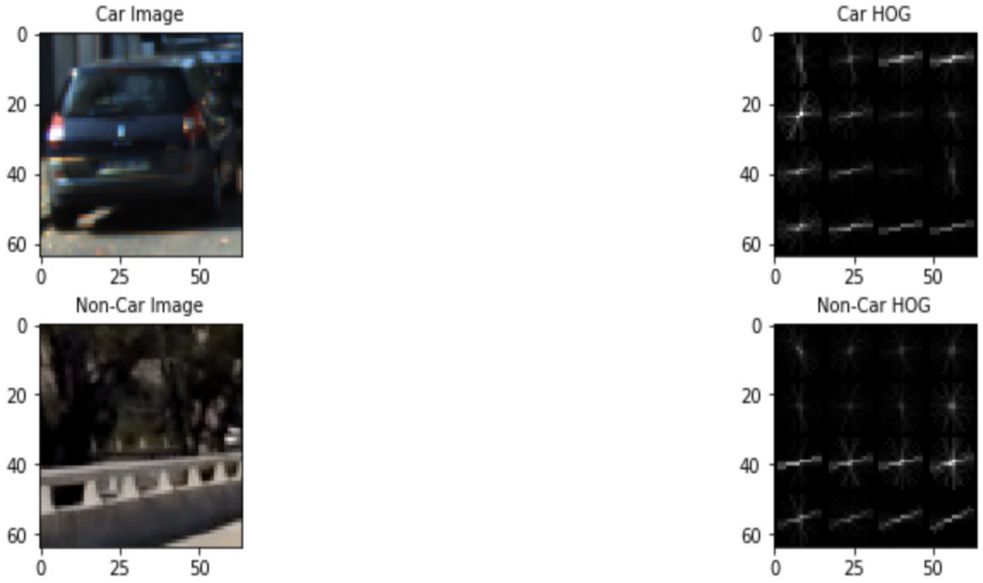
Histogram of Oriented Gradients (HOG)

-The 'get_hog_features' function uses the Scikit-image hog() function to extract HOG features. I experimented with several HOG parameters and an alternate color space as below (A few are documented below):

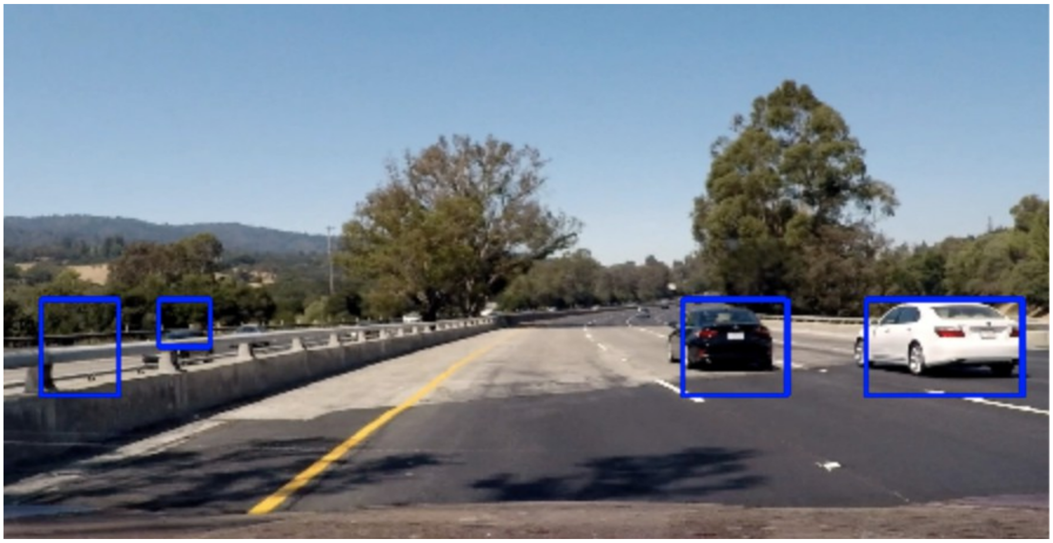
Color Space	**orientations**	**pxels_per_cell**	**cells_per_block**	**Test Accuracy (SVC)**
YCrCb	9	4	1	0.98
YCrCb	19	4	1	0.982
LUV	19	4	1	0.98
YCrCb	19	4	2	0.9842
LUV	5	8	5	0.96
LUV	11	16	2	0.97
YCrCb	**19**	**16**	**2**	**0.9851**

-The 'extract_features' function in turn uses the ' get_hog_features' function to return a list of feature vectors.
I picked the HOG parameters in the last row above after hours of experimentation as I realized that only tweaking the HOG parameters to increase the test accuracy leads to overfitting and does not generalize well to the test images.

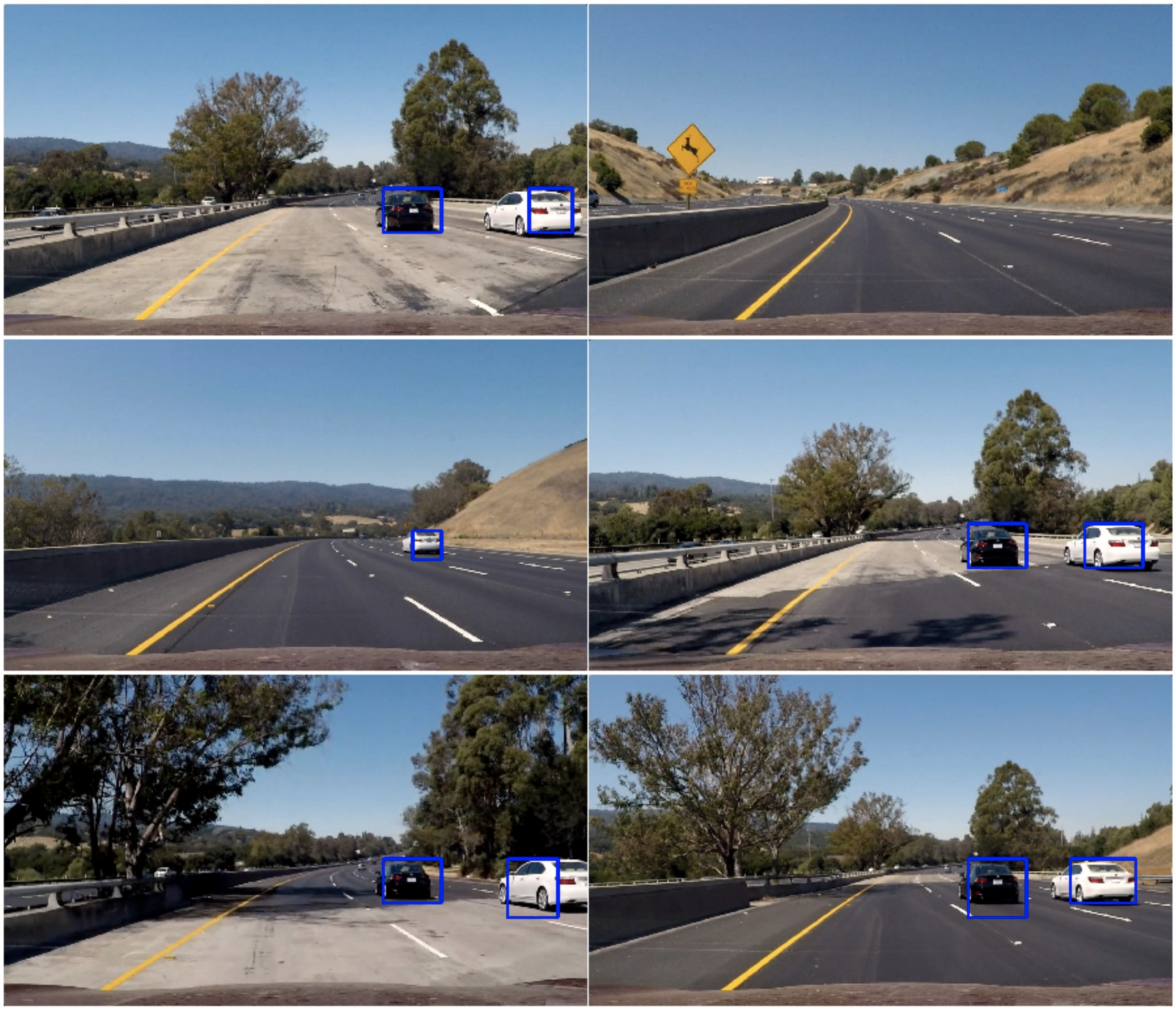
HOG visualized:



For Eg: Even when the Linear Support Vector classifier had a very high (>0.96) accuracy, detection of cars in the test images did not correlate to the accuracy and had a lot of false positives further downstream eg image is as below:



I picked the HOG parameters by keeping the other parameters of the pipeline stable (eg: Heatmap Threshold) and then analyzing the performance on the test images until I got the desired car detection (no false positives, no multiple bounding rectangles) as below:



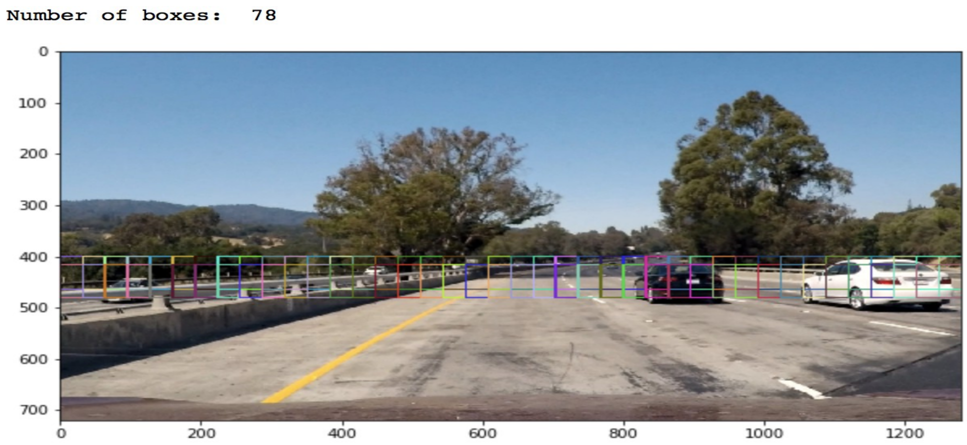
Classifier Training:

-Since the course content suggested that Support Vector Machines work well with HOG features, I picked LinearSVC (SVC with kernel='linear”)
Reference: http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
-sklearn.preprocessing.StandardScaler() was used to normalize the feature vectors for training the linear SVC classifier as was indicated in the projects tips.
-I decided to not use the color histogram and the spatial features.

Sliding Window Search:

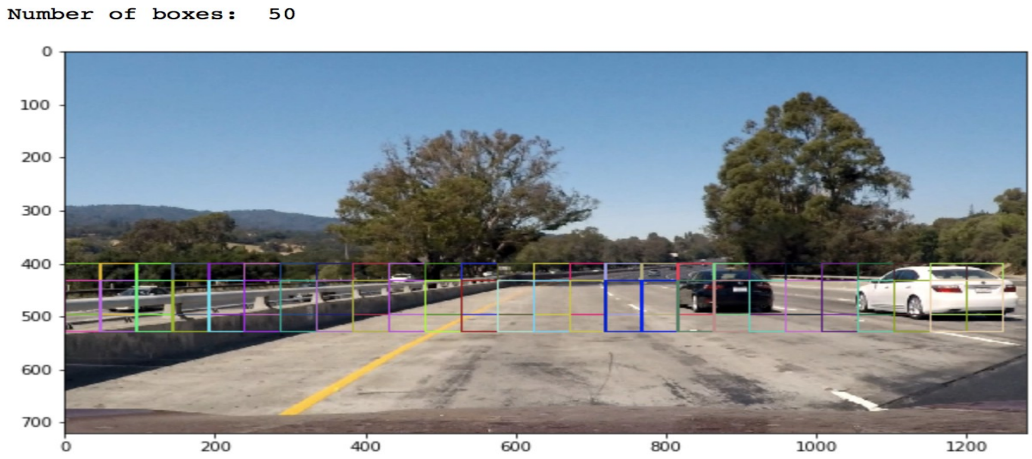
-'find_cars' function was used to for sliding window so that the Hog Features are extracted only once as indicated in the course slide.
-Different scaling values were picked for each run of the function to generate multiple-scaled search windows:

Eg: Scale 1.0,ystart = 416,ystop = 490



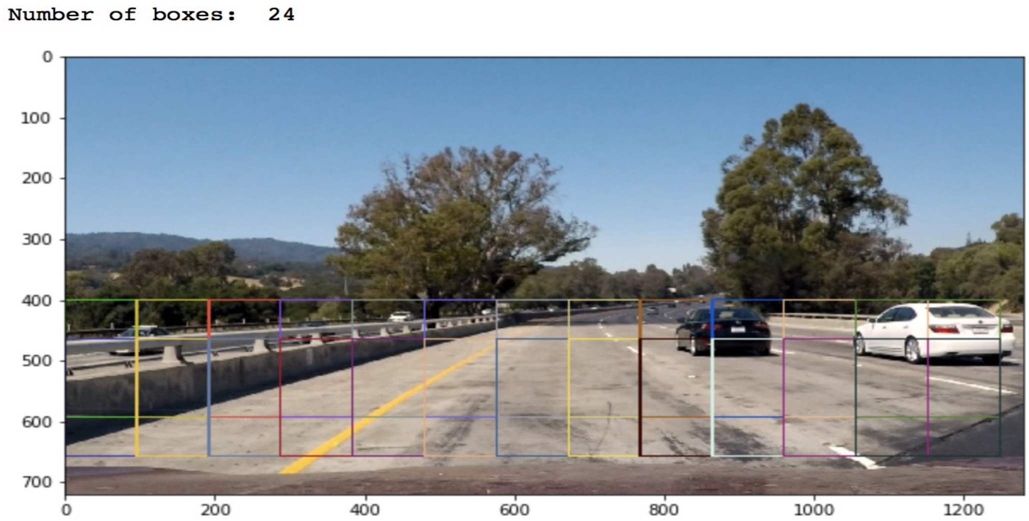
```
Number of boxes:  78
```

Scale: 1.5, ystart = 432,ystop = 528



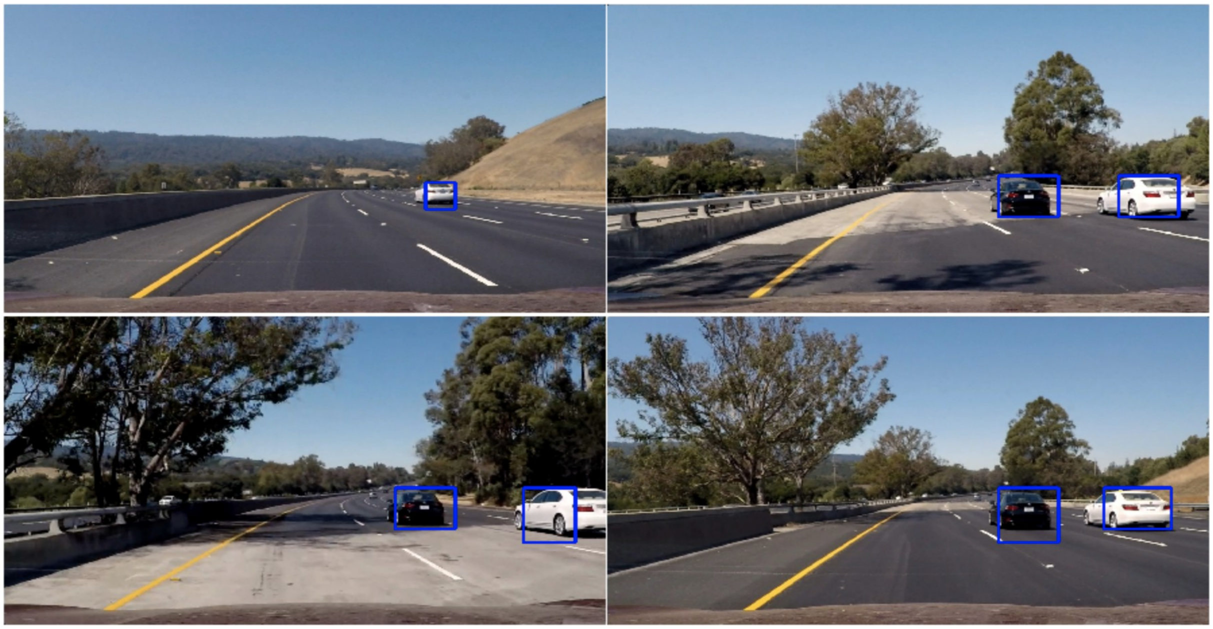
```
Number of boxes:  50
```

Sacle:3.0,ystart = 464,ystop = 660



```
Number of boxes:  24
```

Example of Test Images passed through the pipeline:



How Classifier was improved on:

-We know that cars appear only in certain section of the images. This informed the choice of the ystart values ~(400-660) to exclude the horizon and the hood of the car, beyond this classifier did not need much improvement (discussion of heatmap and thresholding is in the next section)

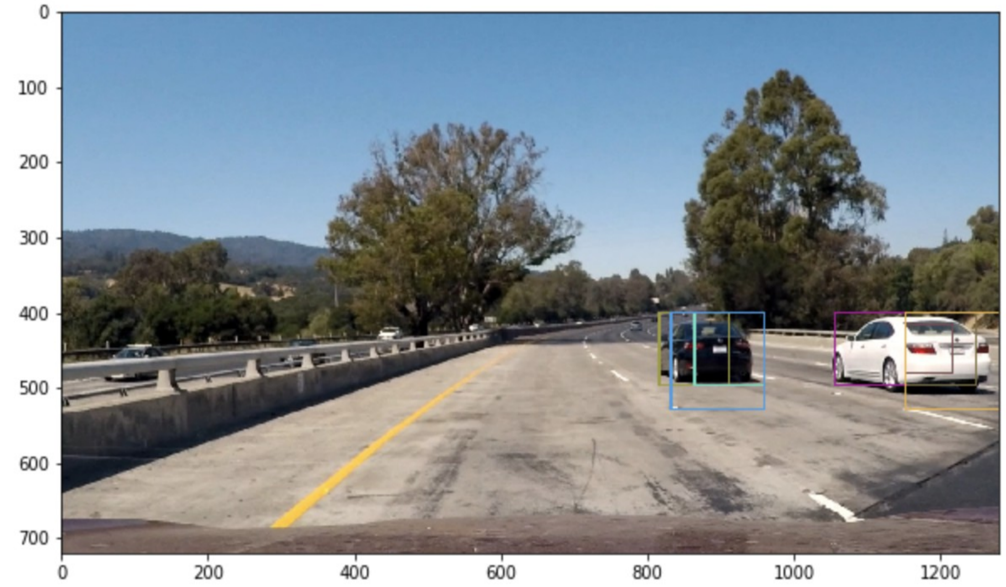
Video Output:

Video output has been uploaded to the repository, I saved prior detections in a buffer and periodically refreshed this buffer as the cars do not move significantly in successive frames which allows smooth translation of the bounding boxes.

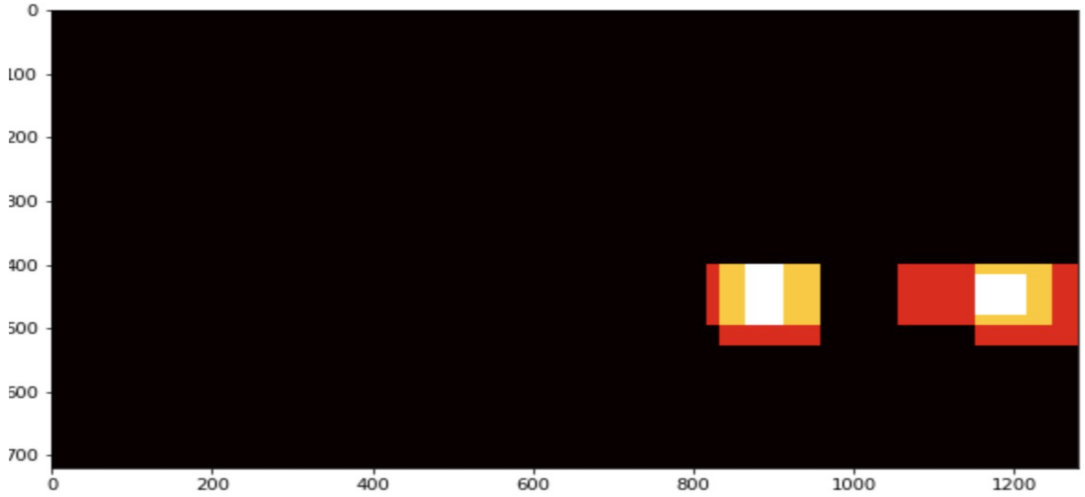
Eliminate Multiple Detections and false positives:

-Multiple detections and false positives were eliminated by building heat maps from the car detections.
-'add_heat' function was used for a list of bounding boxes.

Eg: **Multiple Detections:**

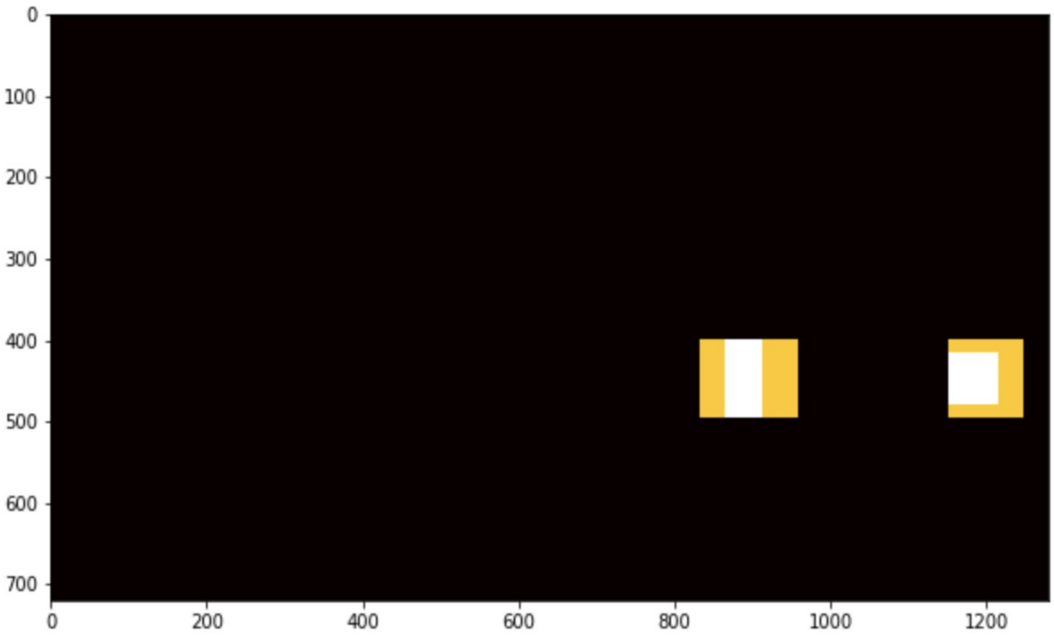


Heatmap:



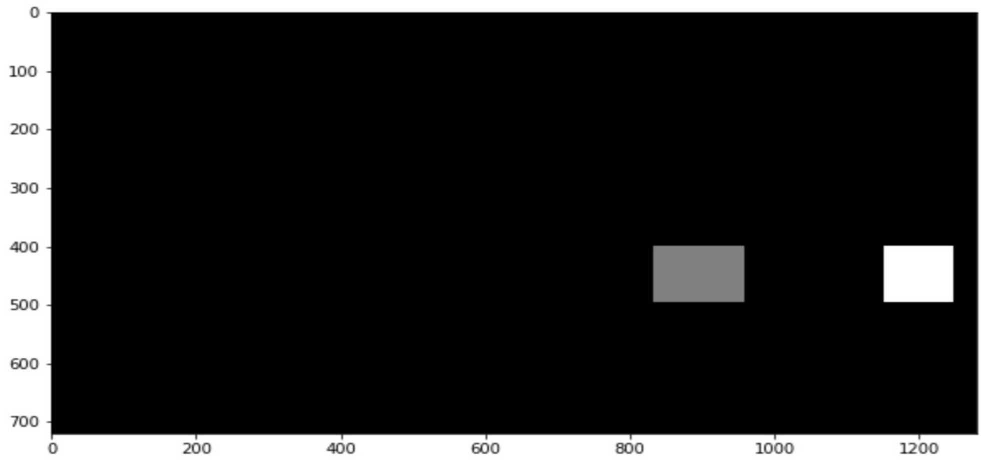
-Threshold was applied (experimented with several values to get the detection right using test images) to the heat map dets

Eg: Heatmap Thresholded:



-label() function from scipy.ndimage.measurements was used as was suggested in the course slide to determine the pixels that belong to the car as below:

```
2 Cars are detected
```



-Function 'draw_bboxes' then draws a bounding box based on the max x and y values.

Problems Faced:
-Picking the appropriate parameters based on one or two test images does not seem to extrapolate to the frames of the test video, I found that running the pipeline on the video and checking how it performed informed by selection of parameters better than overfitting for the test images.
-I encountered several error when trying to add the spatial and histogram features and therefore decided to use only the HOG features, based on the forum posts it seems like the pipeline is working better when only the HOG features are used.

Improvements:
-As I am storing previous detections to smooth out the translation of the bounding boxes, in case of a vehicle which abruptly keeps changing lanes, the bouding box will have a lag to follow the vehicle and might cause multiple detections.