In this project, the goal is to use **computer vision** techniques to detect vehicles on the road.

Steps taken:

· Analysis of data - Car and Non-car images

· Apply an image transform to YCrCb and append binned color features, as well as histograms of color, to the HOG feature vector. This together forms the feature.

· Train a Support Vector Machine classifier

· Implement a sliding-window and use the classifier to search the vehicle

· Generation of Heatmap with detection and bound box on vehicles

· Smooth the results and eliminate false positives in video

Data Analysis:

These images have to be extracted from real world videos and images, and correctly labeled. Udacity provided 8.792 images of car and 8.968 images of non-cars, from sources listed in the attachments. The images have 64 x 64 pixels.

These data are separated in training (90%) and validation sets (10%), and their order is randomized.

**The HOG feature extractor**

The following are the parameters chosen for HOG, Spatial Bins and Color Histograms:-

a) Color Channel - YCrCb

b) Orientation bins - 9

c) Pixels per cell - 8

d) Cells per block - 2

e) Histogram Bins - 32

g) Spatial Image Size - (32, 32)

These parameters are present in cell no. - 5 of vehicle\_detection.ipynb.

### **The classifier**

### **The next step is to train a classifier. It receives the cars / non-cars data transformed with HOG detector, and returns if the sample is or is not a car.**

i) I have trained my classifier using LinearSVC with 10000 random images of car and non-car data sets.

ii) The implementation is present in **cell no. - 7**

### **Applying the classifier in an image frame**

The car can appear in different sizes so different window sizes are required. I have used scales of 1.3, 1.5, 1.8 which corresponds to window of 83, 96 and 115 respectively.

### **7. Smoothing**

One problem of the method described so far is that it detects a lot of false positives: images that are not cars but fool the SVC. The image below is an example of it.

To avoid false positives, we do an average over 10 frames of images. A real car is probable to keep appearing in the image. A false positive will disappear.

### **8. Final video**

Here is the link of final video, also with [Advanced Lane Finding](https://chatbotslife.com/advanced-lane-line-project-7635ddca1960) from previous Project.

Discussion

i) The pipeline used in the current project is very specific to the project video. All the different parameters have been tuned keeping that fact in mind. There is a need for generlistic approach.

ii) Removing False positives was the biggest challenge for me. There are still some False positives in the resultant video. The pipeline needs to be improved further.

The pipeline is not realtime. Deep Learning approach could result in a more general solution and meet real time expectations.