Project 5
Vehicle detection and tracking

The goals / steps of this project are the following:

1. Perform feature extraction on a labeled training set of images, this includes HOG (Histogram of Oriented Gradients) feature and color feature
2. Train SVM classifier
3. Implement a sliding window technique and use the trained classifier to search for vehicles in images
4. Run the tracking pipeline on a video stream. Create a heat map of recurring detections frame by frame to reject outliers and follow detected vehicles
5. Estimate a bounding box for vehicles detected

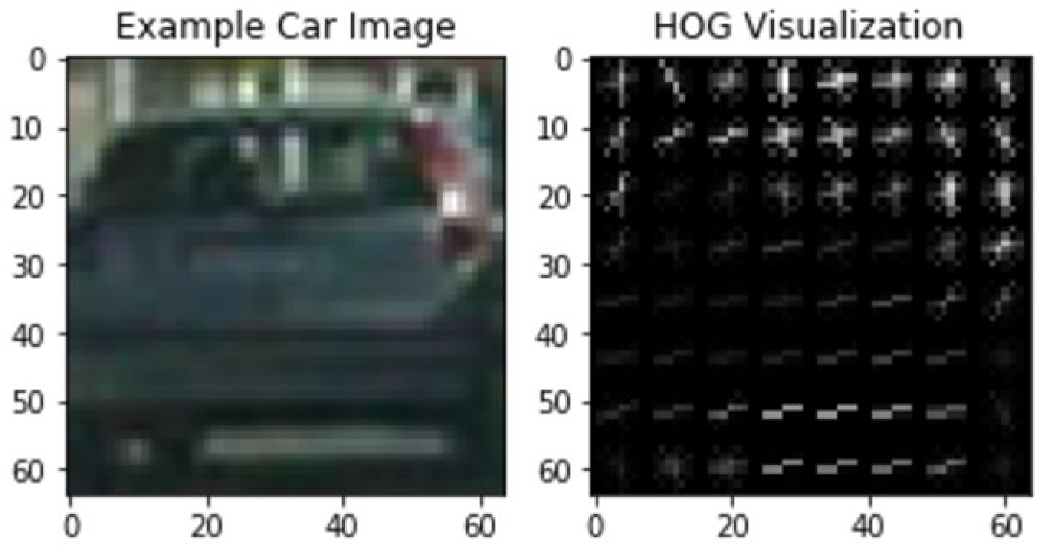
Feature extraction

The following features were extracted from the training data provided.

1) HOG features
2) Color Histogram
3) Spatial bin feature

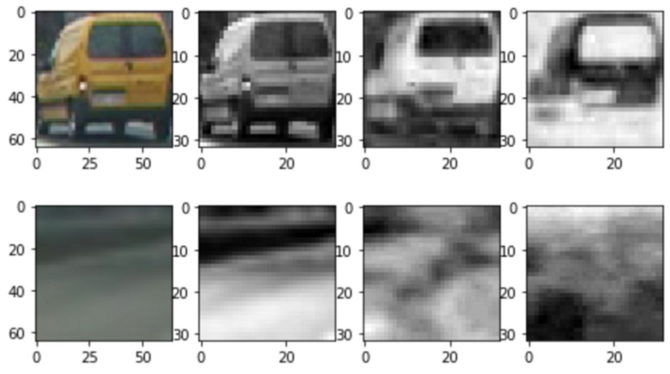
HOG features

For this purpose I have used the images in the training data and applied **skimage.feature.hog** to obtain hog features of an image. Example is as shown below



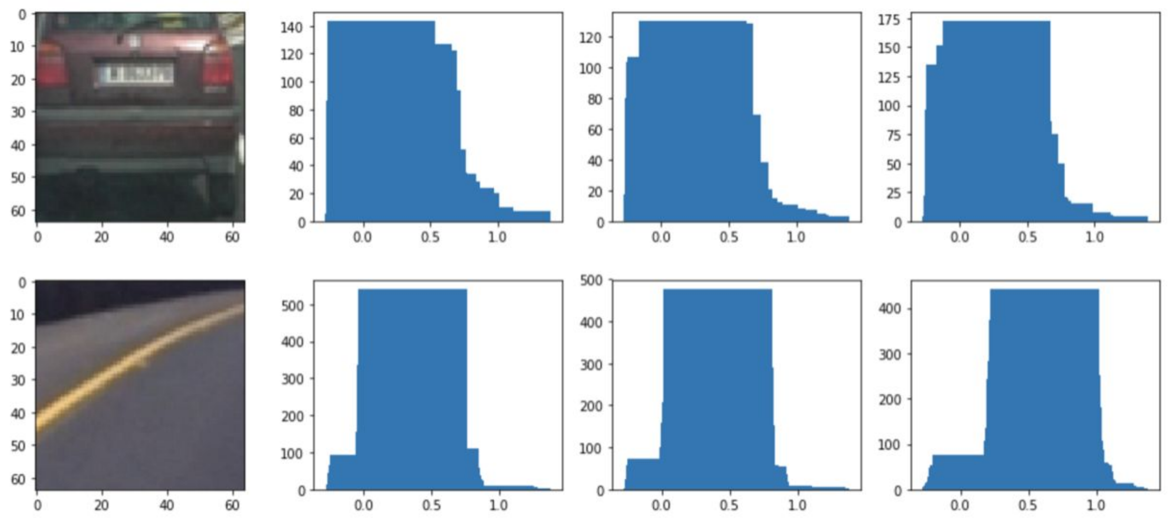
Spatial bin feature

Here we use the image itself as a feature. It is resized to 32x32. The images are converted to YCrCb color space.For this purpose I use the **cv2.resize()** method. Example result is as shown below.



Color Histogram feature

This feature helps us understand distribution of colors in the image. Example result is as below.



Parameters used during feature extraction:

Color space -> YCrCb

Spatial bins -> (32,32)

Hist bins -> 32

HOG features -> orient = 9, pix_per_cell = 32, cell_per_block = 2

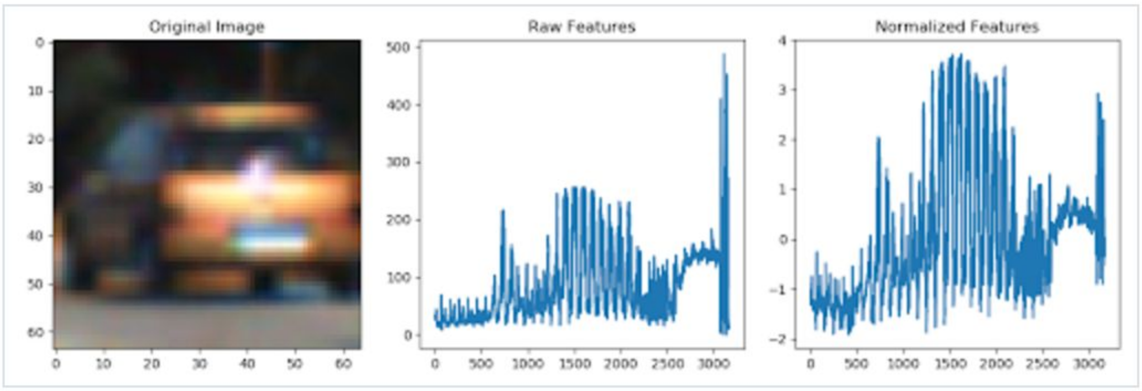
Training a classifier

Classifier used: Linear SVM

Kernel : RBF

Training the classifier took about 1800s. This is because of the large training data.

Step 1: I first obtain spatial bin features, color histogram, HOG features. Concatenating them gives me a feature vector. Then to prevent just one set of features from dominating I normalize the features. Example is as below



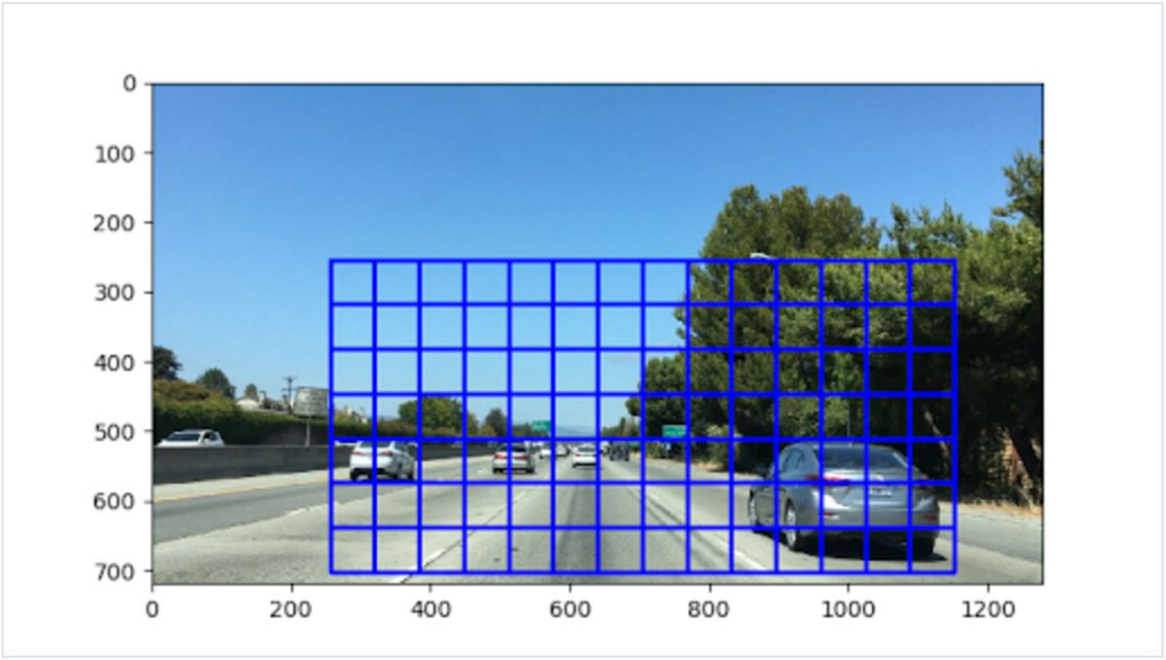
Step 2 : SVM Classifier

Various classifier and features combination tried were as below:

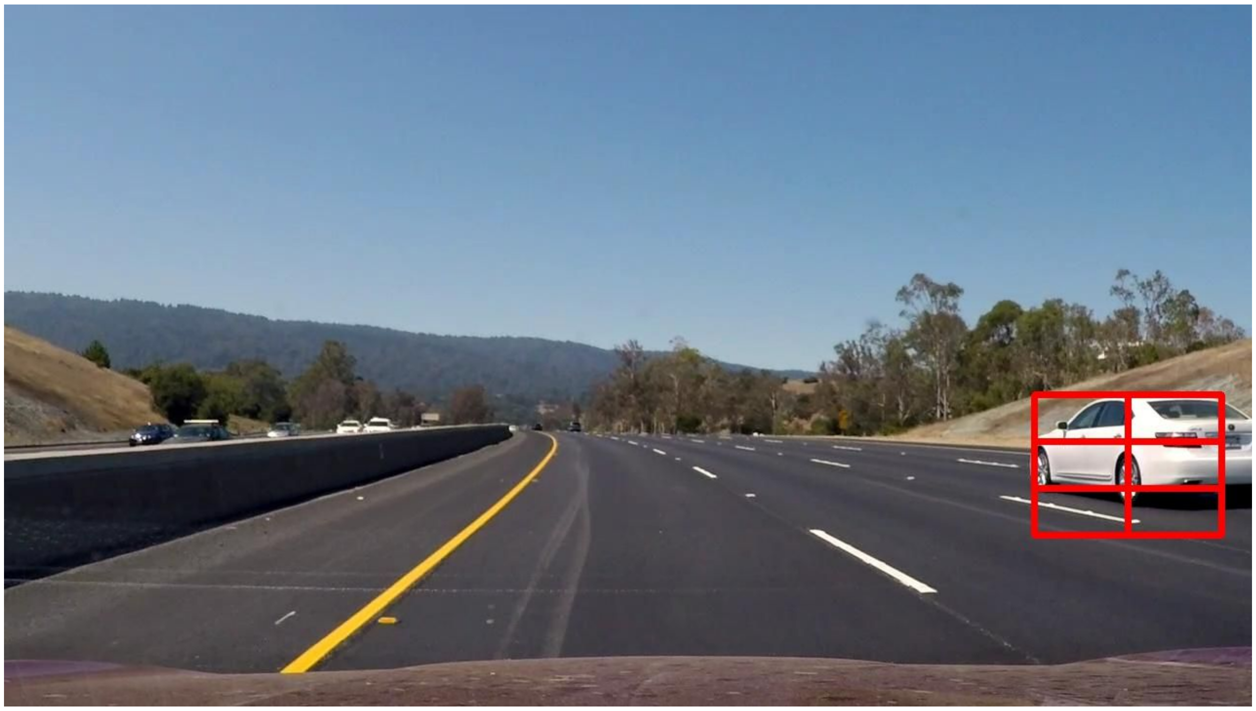
- Linear SVM + HOG features only : This gave me an accuracy of about 92%. This took about 300 seconds to train.
- Linear SVM + HOG + spatial bin: This gave me an accuracy of about 96%. This took about 600 seconds to train.
- Linear SVM + HOG + spatial bin + color histogram: This gave an accuracy of 99.77%. But this took about 1800 seconds to train.

Step 3: Sliding window

Here first we obtain a region of interest where there is a possibility of cars being present. With this we can simply eliminate nearly top half of the image in most cases. So this narrows down our search. Below image shows windows being drawn to emphasize the region of interest.

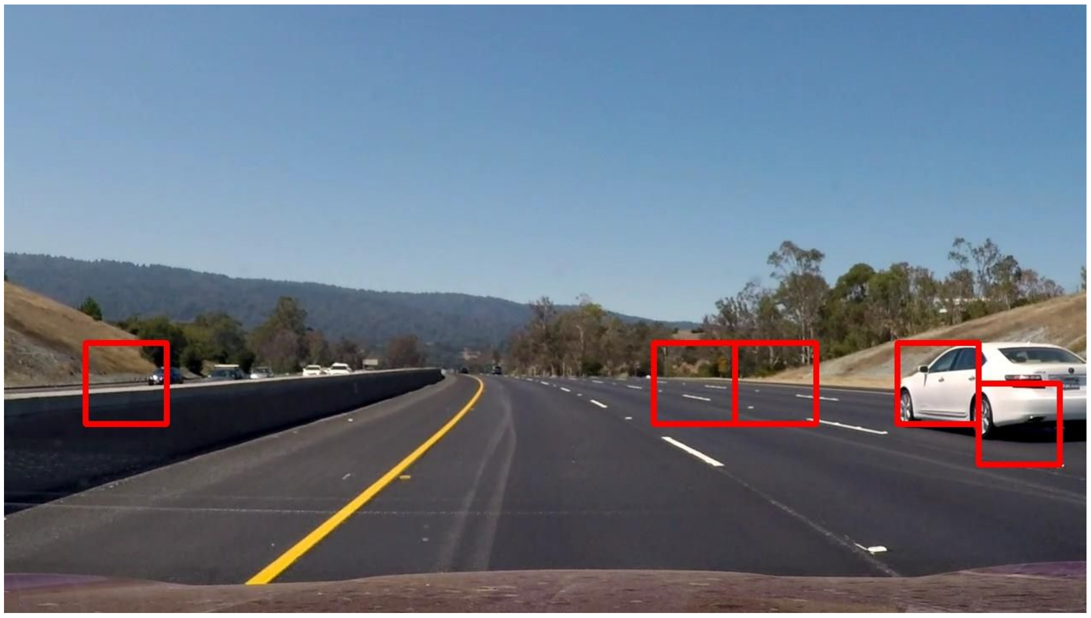


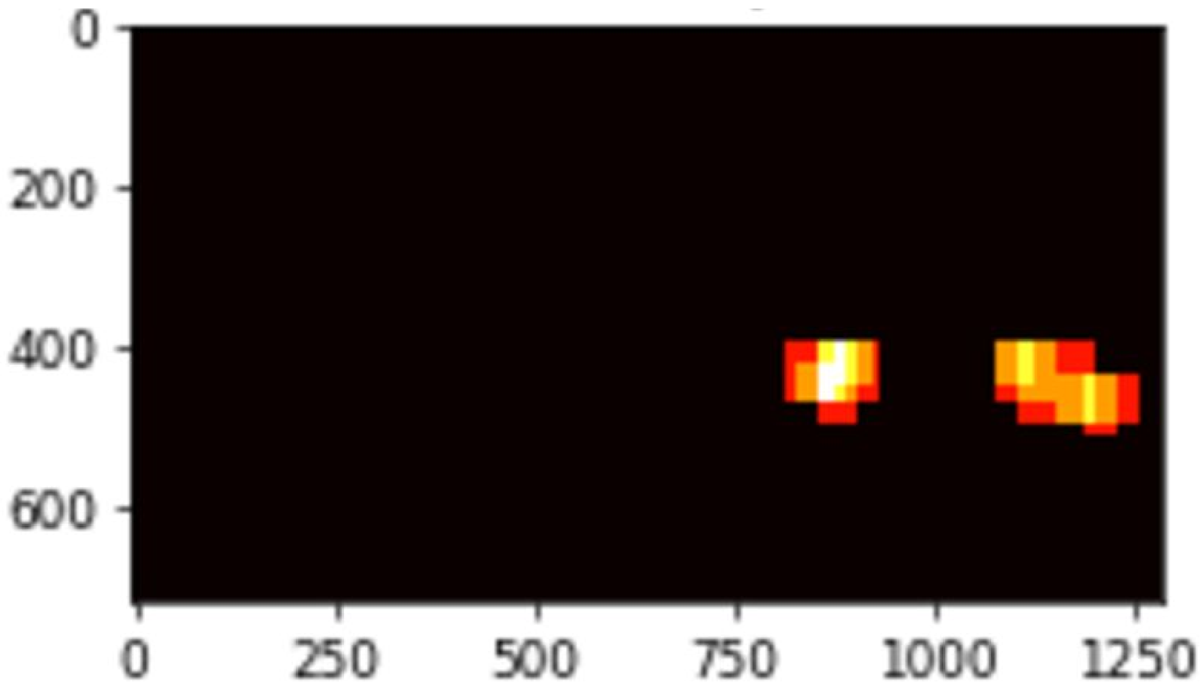
Once we have found the region of interest, we use our trained classifier to predict which windows contains the features we are looking for. Below is an example where classifier finds the features and to indicate this boxes are drawn as shown below.



In some frames we also see false positives. So to get rid of them we use heat maps which help us track down where exactly the cars might be.

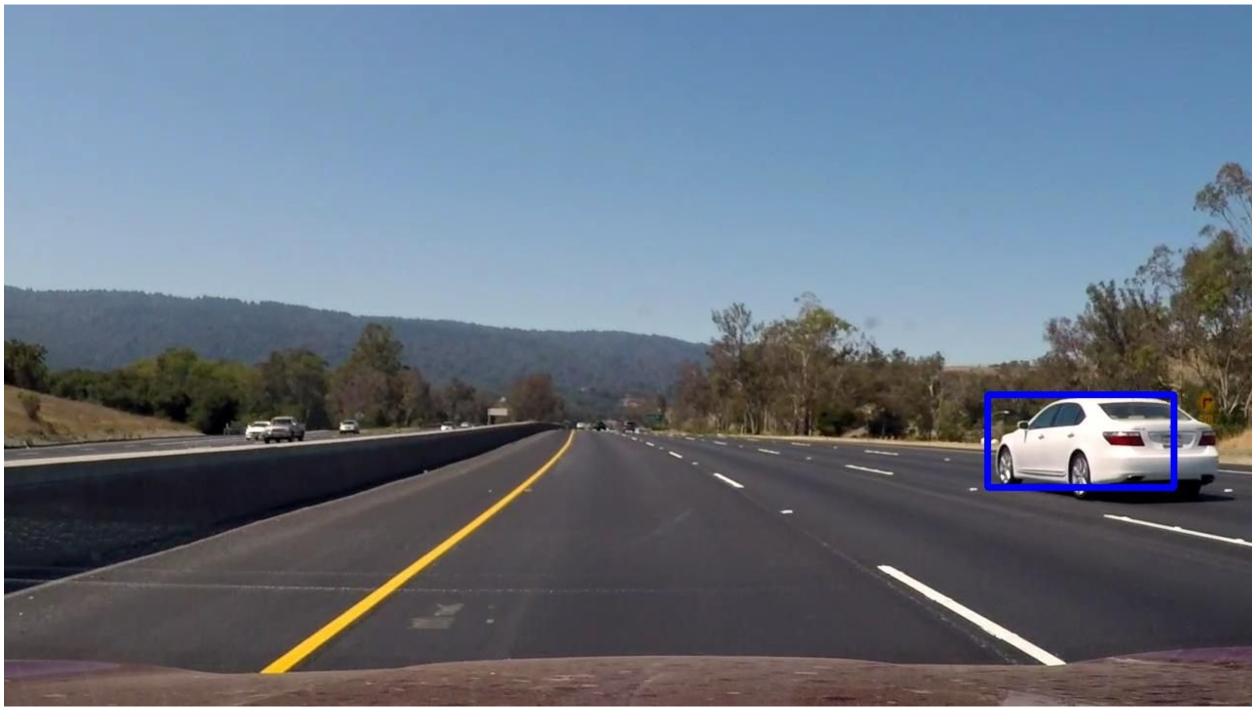
False positive example:

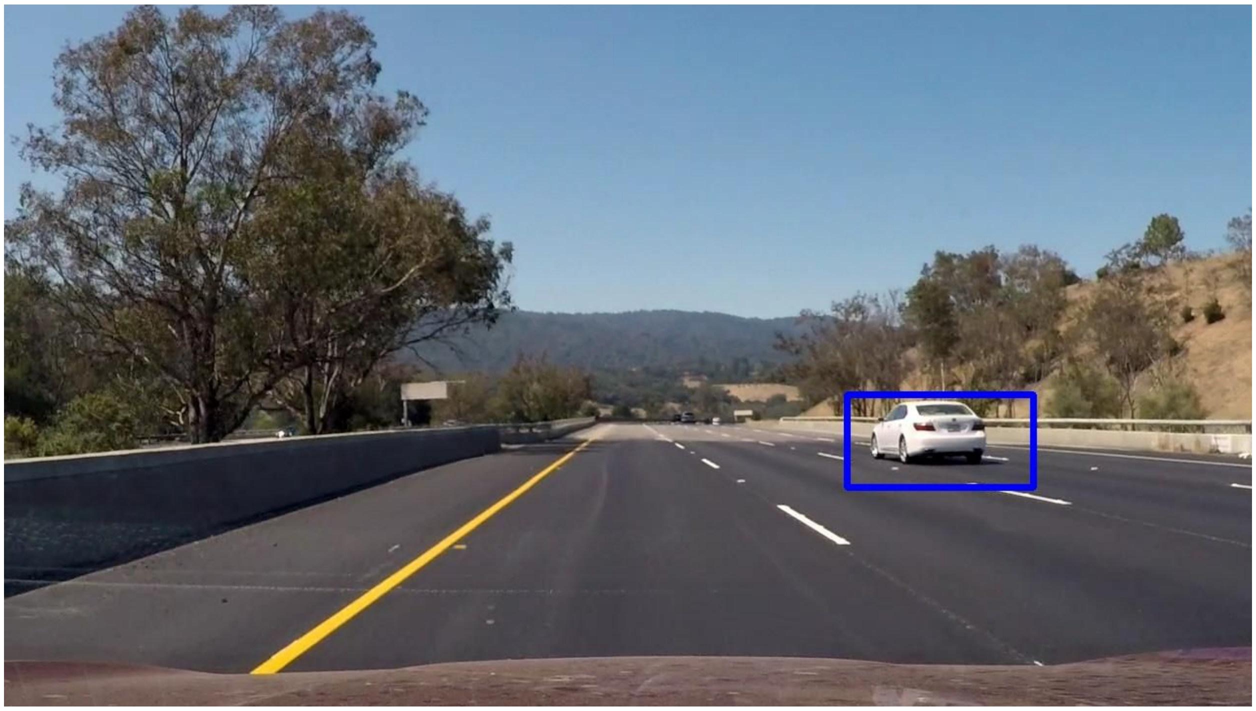




As seen above we now know that there are only two cars and thus eliminating the false positives.

Results:





Discussion:

Problems with pipeline:

- Possible failure causes include , when the car features are larger than the window size. This is the reason the black car is sometimes not detected when it just is visible in the video but is detected when it moves far away.
- The pipeline works well only for this video.
- Need to add more training data to make it more generic.
- White car was not detected with just the data provided originally. I had to add the udacity dataset 1 to training before it started detecting the white car in the video.
- This also fails if the car is in a different orientation, for example this pipeline is not trained to detect cars facing front or side ways.

Possible advances include:

- Using a different classifier to train
- Obtaining more car data so that the pipeline is more generic.
- Obtaining more car data related to video to improve the accuracy of detection
- Average over few frames to make tracking more accurate and faster.