UDACITY

PROJECT

# Vehicle Detection and Tracking

A part of the Self-Driving Car Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

## Meets Specifications

This was an exceptional submission! You have done very well in creating a pipeline that detects the surrounding vehicles at all times, but you could improve on removing false positives as there were one or two in the output video. Keep up the hard work and good luck for the next term!

### Writeup / README

✓

**The writeup / README should include a statement and supporting figures / images that explain how each rubric item was addressed, and specifically where in the code each step was handled.**

### Histogram of Oriented Gradients (HOG)

✓

**Explanation given for methods used to extract HOG features, including which color space was chosen, which HOG parameters (orientations, pixels_per_cell, cells_per_block), and why.**

Good job applying HOG feature extraction and describing how the parameters were chosen. It's great to see that you tried out so many values before selecting the ones that you did.

✓

**The HOG features extracted from the training data have been used to train a classifier, could be SVM, Decision Tree or other. Features should be scaled to zero mean and unit variance before training the classifier.**

The scaling method and classification method have been correctly implemented in code. You could add a little more detail here. Did you try any other model like decision trees or logistic regression? Did you try to tune the parameters for SVM?

You can use GridSearchCV to tune the main parameters like 'C' and 'kernel' (only applicable if you use SVM and not LinearSVC).

### Sliding Window Search

✓

**A sliding window approach has been implemented, where overlapping tiles in each test image are classified as vehicle or non-vehicle. Some justification has been given for the particular implementation chosen.**

Good job describing the approach you have taken. You could consider cropping the image before feeding it to the model or starting the search from an offset on the X-axis (around 400 pixels). This is because the pipeline sometimes detects false positives that are on the other side of the road (oncoming traffic).

Note that using this approach would mean that the pipeline is not as robust, as it would fail if the car was on the right side of the road. Another idea is to use the result from the last frame and search only in this specific region.

✓

**Some discussion is given around how you improved the reliability of the classifier i.e., fewer false positives and more reliable car detections (this could be things like choice of feature vector, thresholding the decision function, hard negative mining etc.)**

Great idea for removing the false positives! You could also try hard negative mining for improving the classifier's performance. For performing this step, you can crop the false detected areas and rescale them by 64 *64 and add them to the non vehicle data .

## Video Implementation

✓

**The sliding-window search plus classifier has been used to search for and identify vehicles in the videos provided. Video output has been generated with detected vehicle positions drawn (bounding boxes, circles, cubes, etc.) on each frame of video.**

The video output is very impressive as the surrounding vehicles are accurately tracked through the entire video and the bounding boxes are also smooth and accurate. There was one false detection but a small amount of error can be expected in such an approach.

✓

**A method, such as requiring that a detection be found at or near the same position in several subsequent frames, (could be a heat map showing the location of repeat detections) is implemented as a means of rejecting false positives, and this demonstrably reduces the number of false positives. Same or similar method used to draw bounding boxes (or circles, cubes, etc.) around high-confidence detections where multiple overlapping detections occur.**

A heat map has been used to reject false positives and bounding boxes are drawn around car images.

## Discussion

✓

**Discussion includes some consideration of problems/issues faced, what could be improved about their algorithm/pipeline, and what hypothetical cases would cause their pipeline to fail.**

Good work discussing the project and suggesting some areas for improvement. You are correct that deep learning approaches like YOLO help to build a more robust pipeline that is not affected by small changes in colors and lighting conditions.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ