UDACITY

‹ Back to Self-Driving Car Engineer

Vehicle Detection and Tracking

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Brilliant Learner,
This project submission was great. It depicts a good understanding of the new concepts introduced in the course. The code implementation and write-up are impressive. Congratulations on completing this last project of SDCND Term 1. I have provided suggestions and additional resources which I believe will help broaden your understanding of the project. Please, feel free to check the links and I urge you to keep working on the project to make it more robust and perfect. Have a nice day and celebrate your achievement👏.

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.

Remarkable writeup. A detailed analysis of the steps taken to produce great results has been provided in the `carnd-vehicle-detection.pdf` file. In addition, techniques such as HOG features and heatmap have been outlined in the report which improves the implemented algorithm. Good work!!.

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.

Impressive explanation in the write up of how the HOG features were extracted. The writeup clearly states HOG parameters used as shown below.

```
Orientations = 31
Pixels per cell = 5
Cell_per_block = 2
```

Advance Learning Tips.

Please, Check out these links below to gain more in-depth knowledge and insight into HOG features,

- scikit-image doc
- Wikipedia _ Histogram of Orientated Gradients.
  I will recommend that you check out OpenCV HOGDescriptor used to extract HOG features which is faster and more efficient compared to Skimage functions.

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.

Commendable work!!. The report clearly states the use of sklearn.svm.LinearSVC in the training of the classifier. Furthermore, Sklearn.preprocessing.StandardScaler was used to normalize the features.

Advance Learning Tips.

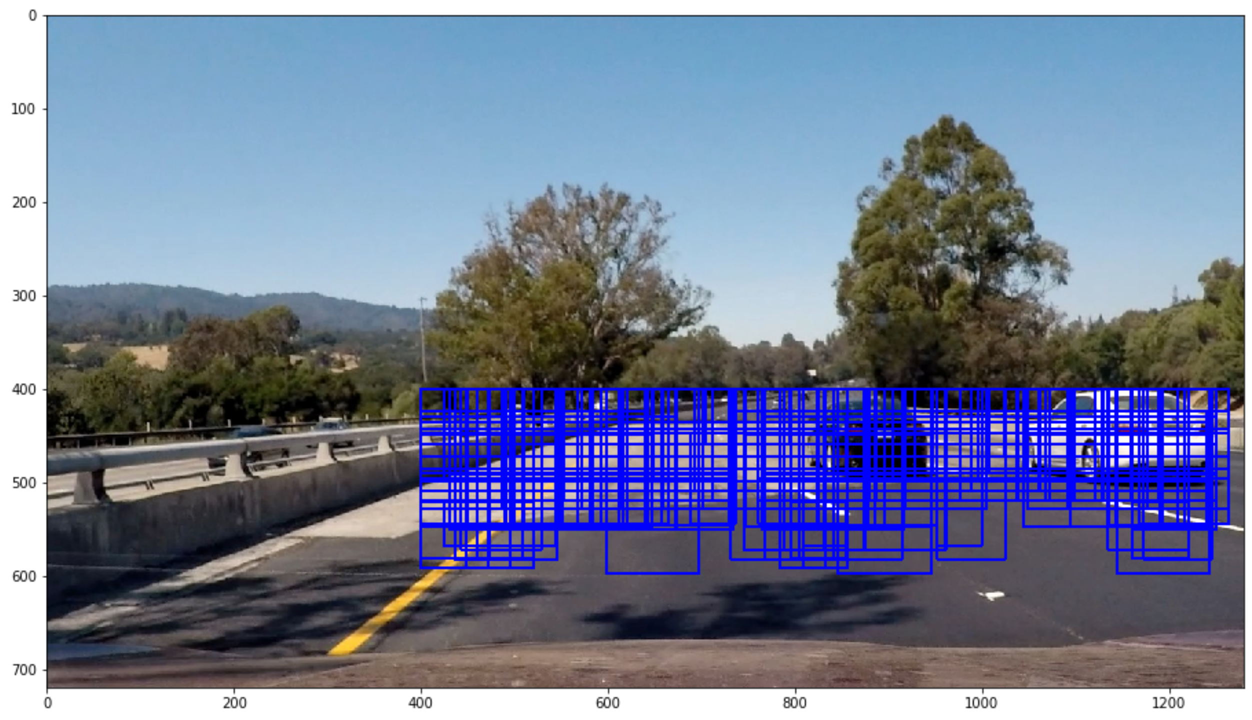
Below, I present a list of alternative classifier which have been proven to produce quite decent results.

- Multi-layer Perceptron Classifier
- Random Forest Classifier
- SGDClassifier
- XGBoost

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.

The project implements the sliding window technique very well and the write-up provides an explicit explanation of how this was achieved. The image below gives a clear visualization of the sliding window technique.



Advance Learning Tips.

More in-depth knowledge can be found in the links below:

- Sliding Windows for Object Detection with Python and OpenCV
- Difference between Sliding Window technique and Scanning Window technique in image searching
- Template Matching

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.)

Splendid work is done on the project to reduce the number of false positives. This was done before moving on to the video processing section. In the submission, heatmap and thresholding techniques were some of the methods implemented to improve the reliability of the classifier among others.

Advance Learning Tips.

Some interesting links on how to improve the reliability of the classifier can be found below.

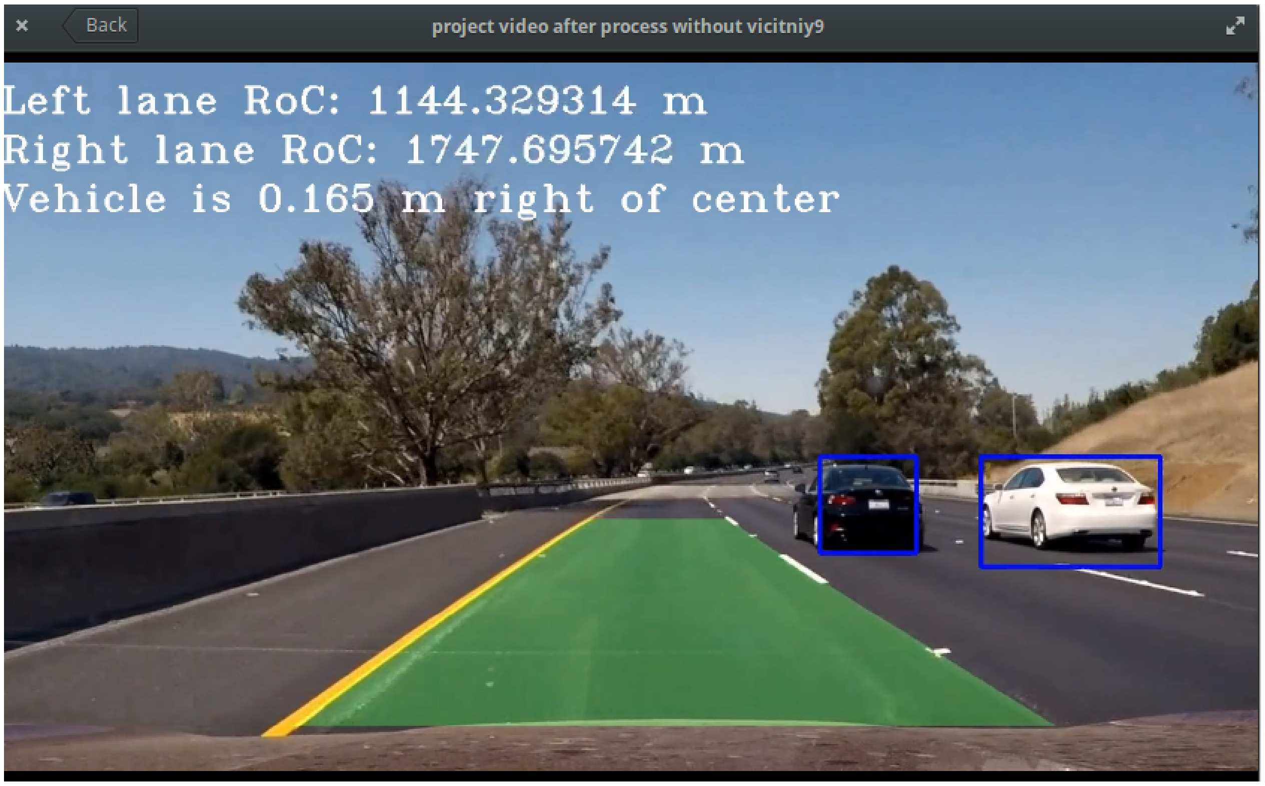
- Hard negative mining could also help improve the reliability of the classifier. This discussion has a down-to-earth explanation of this concept.

- To improve non-linearity of the data provided, one could consider to use random.shuffle() to shuffle the data before passing into model. Here is a great discussion on this subject.

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 good. The classifier together with the sliding-window search has been used logically to identify vehicles in the video and this shown by the bounding boxes in detected vehicle positions.


Back
project video after process without vicitniy9
Left lane RoC: 1144.329314 m
Right lane RoC: 1747.695742 m
Vehicle is 0.165 m right of center

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.

Interesting!!. The implementation of the heatmap and thresholding techniques is great and clearly shows the location of repeated detections on multiple frames.

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.

Great job. The discussion clearly outlines some problems faced and provide reasonable suggestions on how these problems could be addressed

DOWNLOAD PROJECT

RETURN TO PATH