UDACITY

Logout

PROJECT

Vehicle Detection and Tracking

A part of the Self Driving Car Engineer Nanodegree Program

PROJECT REVIEW | CODE REVIEW | NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT

Hello Udacian,
Congratulations! I am very impressed by your first submission. You outputted a visual display of the video provided with a correct detection of vehicle positions drawn with bounding box. Your code is clear, lead to the solution, and you have correctly answered all the questions. You made it.😊

Good luck in the rest of your Self Driving Car Nanodegree Program!

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.

The README looks good, and it includes all the elements required by the specification. It's good because:

THE WRITEUP / README SHOULD INCLUDE A STATEMENT AND SUPPORTING FIGURES / IMAGES THAT EXPLAIN HOW EACH RUBRIC ITEM WAS ADDRESSED

The `writeup.ipynb` a goal of the project and contains 6 images that explain how the rubric item was addressed. First, you provided 3 screenshots of the results and one image of the original , second, an image of the sliding windows, finally an image of the video improvement.

SPECIFICALLY WHERE IN THE CODE EACH STEP WAS HANDLED

The README indicated where the code for each step is located, for example in the HOG section I can read this: **The code that extracts image features is in** `feature_extractor.py`

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.

EXPLANATION GIVEN FOR METHODS USED TO EXTRACT HOG FEATURES

The work is in the right track! The function `extract_features` from the project demonstrates proficiency in using `skimage` histogram of gradients to extract features from images.

DETAILS OF *COLOR SPACE* USED

The writeup includes which color space where use in the following note, *"For color spaces both YUV and YCrCb performed well with YCrCb having a slight edge."*

DETAILS OF *HOG PARAMETERS (ORIENTATIONS, PIXELS_PER_CELL, CELLS_PER_BLOCK)* USED

Great job here that the project is using for the HOG parameters the values provided in the lecture notes, `orientations=9, pixels_per_cell=8 and cells_per_block=2`.

WHY PARAMETER CHOICE WERE MADE

The writeup contents a justification of the method used. I can read here that _"For the HOG parameters the values provided in the lecture notes, `orientations=9, pixels_per_cell=8 and cells_per_block=2` performed well." I can read here that and the selected combination provided over a constant 0.99.3% accuracy which in conjunction with a robust tracking pipeline it provides solid vehicle detection.

Suggestions and Comments:

This link is pretty useful and explains in details what is the Histogram of Oriented Gradient (HOG) feature descriptor, and details behind the algorithm - http://scikit-image.org/docs/dev/auto_examples/plot_hog.html

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.

Great job here using the HOG features extracted from the training data to train the SVM classifier. I am happy to see that the feature values were normalized to value between 0.0 and 1.0 before applying scaling transform to the whole dataset. :)

Sliding Window Search

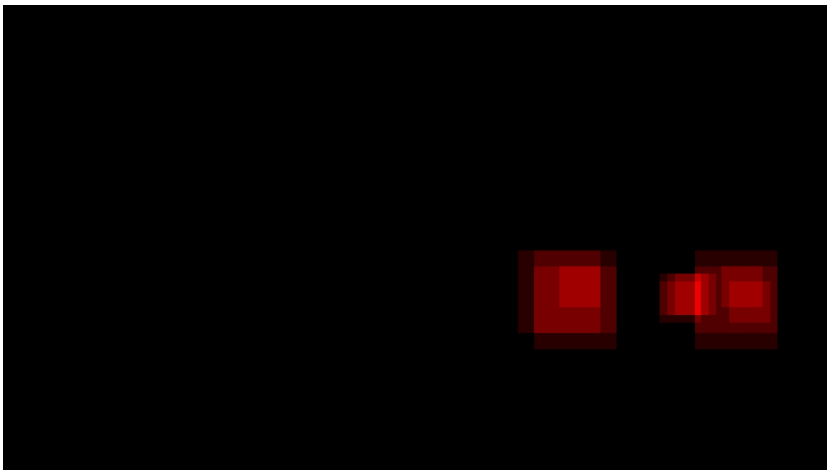
A sliding window approach has been implemented, where overlapping tiles in each test image are classified as vehicle or non-vehicle. Some justification has been given for the particular implementation chosen.

A sliding window approach has been implemented, where overlapping tiles in each test image are classified as vehicle or non-vehicle. A great discussion on how you implemented a sliding window search. There are a few decisions which were made while designing this sliding window search in the `slide_window`. These are all excellent decisions, and leads to a great sliding window.

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.)

SOME DISCUSSION IS GIVEN AROUND HOW YOU IMPROVED THE RELIABILITY OF THE CLASSIFIER

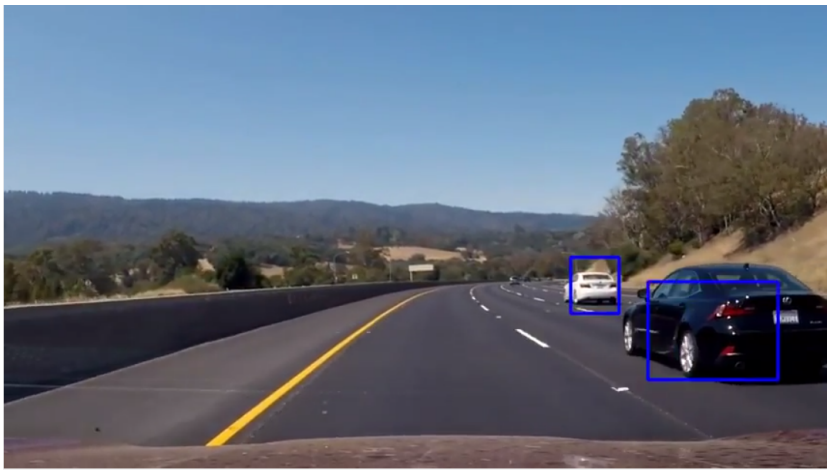
The README provide a great discussion about the improvement of the classifier when it mentioned that:
" In order to reduce false positives we only retained windows that had a prediction probability higher than a threshold value(defaults to 0.8). These windows were used to create a heatmap that was thresholded again to further eliminate false positives." Two example images was provided to show how the improvement work.



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.

A new video with a correct detection of vehicle positions drawn with bounding box was generated.



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.

Evidence for correctly implementing a method for false positive detection is seen from evidence from the blue heat maps in the video output. As required, this greatly reduces the number of false positives.

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.

DISCUSSION INCLUDES SOME CONSIDERATION OF PROBLEMS/ISSUES FACED

The writeup presents some challenges encountered in the discussion section. Namely, *experimenting with various parameters to make the project work*.

WHAT COULD BE IMPROVED ABOUT THEIR ALGORITHM/PIPELINE

A great discussion on what could be improved in the current pipeline was made when it mentions that, *"I believe using something like U-Net or YOLO can provide better results and this is what I intend to experiment with next."*

WHAT HYPOTHETICAL CASES WOULD CAUSE THEIR PIPELINE TO FAIL

There is no case of drawback in the current pipeline that was discussed explicitly. To improve this work a discussion can be on the cases where the current pipeline can fails such as:

- city conditions,
- mountainy road,
- or other real condition

DOWNLOAD PROJECT

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

RETURN TO PATH

Student FAQ