Capstone Project

Alberto Rivera
January 23, 2016

Machine Learning Engineer Nanodegree

Domain Background

This project is inspired by the Vehicle Detection and Tracking project of the Self Driving Car Nanodegree offered through Udacity. This project focuses on image classification as a subdomain of the field of Computer Vision. More specifically identifying cars in a set of images.

Problem Statement

The problem at hand is the identification and classification of vehicles in an image. The solution to the problem should be able to segment relevant objects in an image in order to accurately identify a vehicle in a image. As for quantification of the proposed solution we could measure the accuracy of the classifier with the training and testing data as well as the percentage of vehicles classified correctly in a given number of images. The project and results should be replicable using the same image dataset.

Datasets and Inputs

For this project we will use a dataset consisting of labeled car and non-car images that will be used to train our classifier. Once the classifier is trained it will be used on a set of images which will act as our input and will result in a video. The image data set and the input video can be found at the following locations:

- Vehicle images
  https://s3.amazonaws.com/udacity-sdc/Vehicle_Tracking/vehicles.zip

- Non-vehicle images:
  https://s3.amazonaws.com/udacity-sdc/Vehicle_Tracking/non-vehicles.zip

- Source video:
  https://github.com/udacity/CarND-Advanced-Lane-Lines/blob/master/project_video.mp4

Solution Statement

The problem of identifying vehicles in a source video will be solved by using computer vision and machine learning techniques. Specifically a SVM classifier will be trained using the labeled image dataset, which will then be applied to the source video. Once the vehicles have been identified on each frame a detection box will be drawn around it. The result will be a copy of the original video that has bounding boxes drawn on the vehicles that the classifier was able to identify. To measure the success of the algorithm we will use the rate of detection. We know how many vehicles are in each frame of the video, so we will use the percentage of correctly identified pedestrians for each frame for the duration of the video.

Benchmark Model

As a benchmark model we will use a report by Old Dominion University titled "Exploring Image-Based Classification to Detect Vehicle Make and Model" This report uses a algorithm pipeline very similar to the one proposed for this project. Hover the object of this report is slightly different. This report intends to identify different types of vehicles (cars, motorcycles, trucks, etc.) on a source video which is captured from a fixed location. In our case we are only identifying cars and our source video was captured from a moving vehicle itself. While the report used as a benchmark model is different to the project described in this proposal, the methodology is similar and therefore the results for the performance of the algorithms are comparable.

The benchmark report can be found here:
https://www.uidaho.edu/~/media/UIdaho-Responsive/Files/engr/research/tranlive/reports/ODU%20-%20Image%20Based%20Vehicle%20Classification.ashx

Evaluation Metrics

In order to evaluate the performance of our algorithm, two metrics will be used. The first will be the accuracy of the trained support vector machine classifier on the training and testing data. The second metric will be used to quantify the final result of the algorithm. This requires manually identifying cars in the source video and comparing the number of manually identified cars to the number of cars identified by the algorithm.

Project Design

The design of this project will follow an incremental approach divided into seven parts, each building on the previous parts.

- Part 1: HOG Feature Extraction
    - The first part showcases the implementation of a Histogram of Gradients (HOG) Feature extraction. We will use a test image as input in order to demonstrate how it can be converted into a set of features and show a visual representation of the features.
    - Then the same feature extraction will be done for all the images in the data set.

- Part 2: Train a Linear SVM classifier
    - Using the features extracted from the car and non-car datasets in the previous part we will train a Support Vector Machine (SVM) classifier. Then the accuracy of the classifier will be tested on the training and testing data.

- Part 3: Add Color Features
    - In this part further features will be extracted. In the features come from the color of the image rather than the gradient. Features from all of the images in the dataset will be extracted and combined with the HOG features.

- Part 4: Re-Train the Linear SVM classifier
    - The classifier from Part 2 will now be retrained with the combined HOG and Color features. Once again the accuracy of the classifier will be tested on the training and testing data.

- Part 5: Sliding Window Search
    - This part focuses on segmenting a large image into smaller more practical images in order to search for vehicles. This will result in a set of windows from which images will be extracted to then be classified.

- Part 6: Search and Classify an Image
    - Here all the previous parts come together to achieve vehicle detection in a single image. This image will be a single frame of the “dash cam” source video. A sliding window search will be used to break down the image into more manageable images than will then be resized in order to be classified properly. If the classifier predicts that there is in fact a car in the image a bounding box will be drawn around the car.

- Part 7: Search and Classify a Video
    - This last part is very similar to Part 6, with the exception that the the search and classification will be applied to each frame of the source video and the result will be a copy of the source video but with the vehicles identified by bounding boxes.