1. Introduction

In assignment #03, we introduce our approach on creating a car detector for the final report. Similar as in assignment #02 with the eye dataset, we are going to separate the problem into two different components, the car detector and the car classifier. The car detector is responsible for finding potential regions of interest, where a car might be in the image using different object detection techniques. The car classifier will then be applied to the regions of interest detected by the car detector. The car classifier will use a supervised learning algorithm to detect whether a given image patch is a car or not.

2. Methods

Here we go into detail about the exact methods we want to apply for our car detector and our car classifier.

2.1. Car Detector

The car detector will analyze the input image and find regions of interest for further processing in the car classifier. We want the car detector and car classifier process to be independent from each other. This allows us to easily exchange the chosen classification method in the classifier without having to change the car detector.

The car detector will use image pyramids to find cars of different sizes. In image pyramids the image is down sampled to a lower resolution, so that when analyzing a window of a fixed size, we can detect a full car, that was previously too large to be completely visible in the window.

We are also going to apply a sliding window, so that we can detect cars regardless of their location in the image. A window of a fixed dimension will slide over our region of interest and pass each image patch to the car classifier. The window size in our car detector will be 64x64 pixels to correspond exactly with our training data of our car classifier.

We expect many possibilities on optimizing the speed of the car detector by significantly reducing the number of image patches that are passed to the car classifier in each video frame. The goal is to create a car detector that can run smoothly when given a video with multiple cars to detect in every frame.

2.2. Car Classifier

The car classifier model will be based on a supervised learning algorithm. The model will be trained on a set of car and non-car images of the size 64x64 pixels. The initial set of positive car images has only 375 images in total. We want to enlarge this positive set by applying different data augmentation techniques. This will include inverting the image along its 2nd (y) axis, applying a random translation to the image, and applying a (small) rotation of the image. We expect this to increase the performance of our car classifier and make it more robust.

We intend to experiment with different supervised learning techniques for the final report and compare their accuracy. The following algorithms we intend to test:

- A support vector machine (short: SVM) based on the HOG features detected in the input image. This will be like our implementation of the pedestrian detector in Exercise 10, but applied on a different problem domain.
- A convolutional neural network (short: CNN) based on the raw input data. Deep learning implementations like CNNs often require a large set of test samples, so this will require us to

enlarge our dataset to successfully train our CNN. Our neural network will not include too many hidden layers simply based on the small GPUs of our laptops we intend to use for training.

We are interested in seeing how well a complex model like CNN with only raw image input performs against a simpler model (SVM) with a more sophisticated set of features (HOG).

3. Outlook

We expect our final report to include an efficient implementation of a car detector and a car classifier with a high accuracy. The report should also include a comparison of different machine learning algorithms and their performance for the given problem domain. We expect the best solution to most likely include both a more sophisticated machine learning algorithm, as well as a substantial amount of feature engineering.