Assignment #03

Introduction to Image Analysis

and Machine Learning

Development of a car detector

based on a supervised learning approach.

Group M15

Emanuel Alogna (emal@itu.dk)

Peter Jan Mortimer (pjan@itu.dk)

**1. Introduction**

For our final report, we developed a car detector using a binary classifier. Our goal was to develop a car detector that can reliably detect cars in videos. Our application can be split up into two major components: the car classifier and the car detector.

In the car classifier component, we train a binary classifier based on a dataset of “car” and “non-car” images. In this assignment we experimented with different machine learning algorithms and compared their performance on this task.

In the car detector component, we use different image processing steps to prepare images in the test videos for our car classifier component. We used different techniques to improve the performance of the car detector to increase the detection speed of our application.

**2. Methods**

In this chapter, we introduce the two components of our application: the car classifier and the car detector.

**2.1. Car Classifier**

Our car classifier uses a binary classifier to determine, whether the given image resembles a car or not. There are different image preprocessing steps depending on the chosen classification algorithm.

**2.1.1. Dataset**

The original assignment dataset includes 5 videos of a highway. 375 64x64 images of cars on the highway were manually extracted. Frames from the videos that do not include any cars are used to generate 64x64 image patches as negative samples for the dataset.

We only want to determine whether a given image includes a car or not. Our data only contains the labels , where labels a non-car sample and a car sample.

**2.1.2. Data Augmentation**

We extended our dataset using different data augmentation approaches. To reliably classify car images from different perspectives and lighting conditions requires a larger dataset. This can be done without manually extending the car images, which could be done by manually annotating videos. The dataset is extended by using different image processing techniques on the existing samples.

We increased the number of positive samples by applying a random rotation of degrees to each car image and reflecting the image along its center y-axis (see Fig. 1). We believe this data augmentation step to be necessary to increase the number of samples to train more complex models, like convolutional neural networks.



Figure 1: Data Augmentation applied to one example car image. Image (a) show the original car. In image (b) the image is reflected along the center y-axis. In image (c) the image is randomly rotated and reshaped to the 64x64 input resolution.

**2.1.3. Histogram of oriented Gradients**

The histogram of oriented gradients (short: HOG) has proven to be a reliable descriptor to detect pedestrians in images. The HOG features are based on counts of the image gradient for local patches within the chosen image. For one 64x64 image sample we produce a HOG vector with 1764 features.

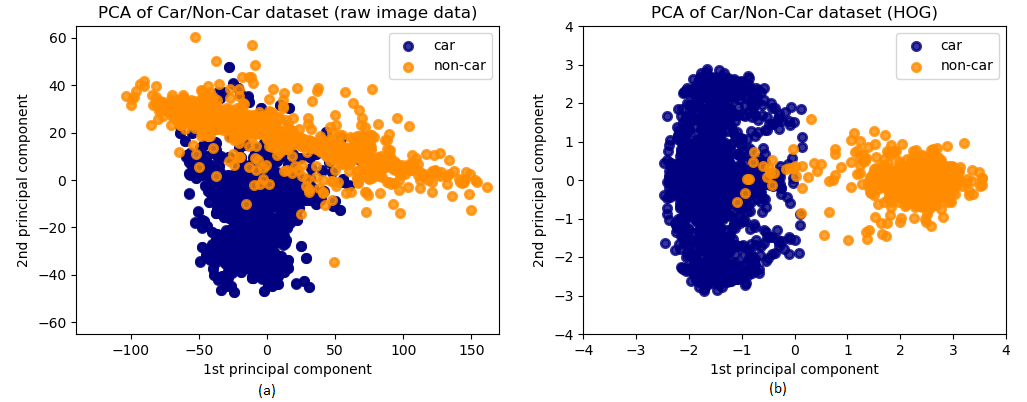
The effectiveness of the HOG features in separating between the two class labels car and non-car can be visualized using principal component analysis (short: PCA). PCA is a method to reduce the dimensionality of our dataset by applying a linear transformation to our dataset. The dataset is transformed to new axes, the principal components. Taking the first two principal components and plotting them can give us a good understanding of the basic structure of our data. In Fig. 2, we calculated the first two principal components of the raw grayscale image from each of our images and calculated the first two principal components of the HOG vector for each of our images. The HOG feature vectors show a better separation of our class labels in 2 dimensions than the raw image data does. This suggests that a model will calculate a better decision boundary for our classes using hog features than just the raw image information.

Figure 2: Here are the first two principal components plotted for the raw grayscale image data (a) and for the calculated HOG features (b). The HOG feature vectors show a greater separation between the image classes. This suggests that a classifier model will be better at determining a good decision boundary when learning using the hog features than the raw image data.

A closer analysis of our HOG feature vector plot shows an overlap for some non-car instances with the cluster of car instances (see Fig.3a). These overlapping non-car instances are images of the road surface marking or other street signs (see Fig.3b). These types of images we expected to resurface as possible false positives when evaluating our car classifier.

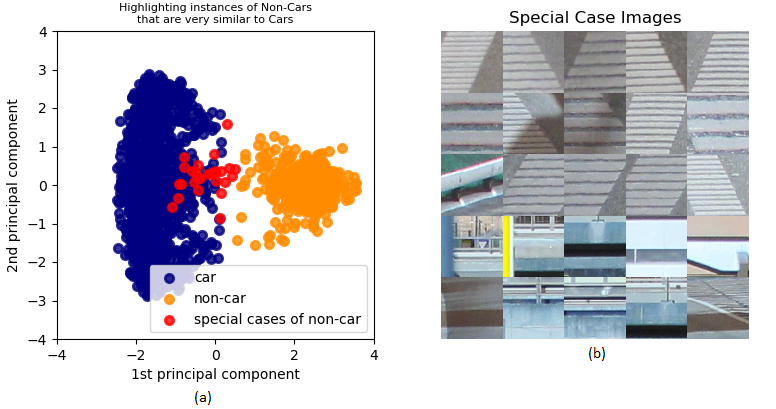


Figure 3: The read marked points in the plot of image (a) are non-car instances, that show similar HOG vector properties as car instances. In image (b) we display 25 of these special non-car instances and see that most of these images resemble road surface markings or street signs. We expect our classifiers to more likely detect false positives around road surface markings based on this observation.

**4. Support Vector Machine**

**2.1.5. Convolutional Neural Network**

**2.2. Car Detector**

**2.2.1. Background Detector**

cv2.createBackgroundSubtractorMOG2()

**2.2.2. Image Pyramids and Sliding Window**

**2.2.3. Overlapping Bounding Box**

**(2.2.4. Predicting Car Destination)**

**3. Results**

**3.1. Model Comparison** (SVM vs. CNN)

**3.2. Car Detector Performance**

**4. Discussion**

**4.1. Improving the Car Classifier**

**4.2. Improving the Car Detector**

**5. Conclusions**

**5.1. Comparing SVM and CNN**

**5.2. Extending the Application**