Vehicle detection in dashcam videos - Report

Antoine DARGIER, CentraleSupélec, MDS-SDI

March 15, 2023

1 Introduction

In this challenge, the goal is to detect vehicles from dashcam videos. We have a first set of 2,222 labelled data, with bounding boxes to spot where cars are. And we have a test set, with 202 pictures where we need to find cars.

In the first test, I tried to use the Viola-Jones algorithm. It is based on Haar-like features, and classifiers are trained for specific tasks. By finding pretrained classifiers, I tested this method and obtained a Sørensen-Dice coefficient of 22%. Another method, which we will detail here, is based on the Histogram of Oriented Gradients (HOG) and a Support Vector Machine (SVM) classifier allowed me to reach 40%. I will describe and comment on this method.

2 Data Preparation

Whatever methods are used, training data is needed, divided into positive examples (here cars), and negative ones (other elements). In order to gain accuracy, I wanted to create my own data using the training data provided. First of all, keeping in mind that the test data was during the day, I only kept the train data when it was daytime. I created positive examples using the bounding boxes. For the negative examples, I went through the photos looking for an area that did not intersect with a bounding box. This gave me 4,363 photos of cars, and 4,363 photos without cars, in 64x64 format. But, in the end, the results were not so good with a lot of elements detected in pictures and not only cars. So I decided to use a vehicle dataset, coming from a Kaggle dataset for Vehicle Detection [1]. The dataset is composed of 17,760 images, 8,792 images of cars and 8,968 images of non-vehicles, all in 64x64 format. Here is an overview:

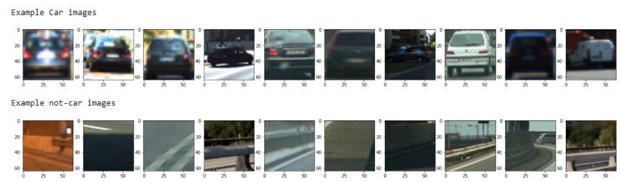

Figure 1: Examples of cars and non-cars sets

3 HOG Features

Histogram of Oriented Gradients (HOG) is a well-known method to detect objects, focusing on the shape of an object. The method takes portions of the image and counts occurrences of gradient orientation. It then generates 9-point histograms with the magnitude and orientations of the gradient, with 8 pixels per cell and 2 cells per block. These are traditional values for HOG features, that you can find here [2]. After some normalization and concatenation, we obtain such figures:

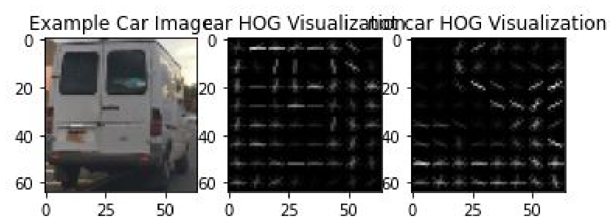

Figure 2: Gradient magnitude visualization

4 SVM

Support Vector Machines (SVM) is a popular machine learning algorithm widely used in object detection due to its ability to handle non-linearly separable data and its ability to generalize well to unseen data. It is also robust to noise and outliers in the data. In addition, SVM combined with other techniques, such as HOG feature extraction and sliding window search, can achieve state-of-the-art performance in object detection tasks. SVM is also computationally efficient and can handle large datasets, making it a popular choice for object detection in real-world applications. That is why I choose this classifier.

The extraction of HOG features is quite long (700s, about 12 minutes), and the training of the SVM takes 45s. But the results are very good, with an accuracy of 99%. Our classifier is now able to detect cars or not!

5 Parameters to play with

To perform the best results, there are a few parameters to fine-tune. First, we can define a zone of interest and look for cars only in this area. That's interesting because it allows us to look neither at the car's dashboard nor at the sky. Then we have to define the sizes of sliding windows, that scour the image for vehicles. In the foreground, vehicles are larger so it is important to set large windows, whereas cars in the distance are smaller and therefore need to be detected by small windows. I found that 3 window sizes gave the best results. Here is the mapping created:

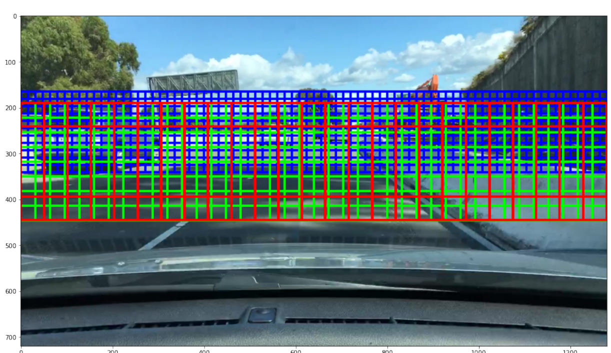

Figure 3: Sliding windows of three scales

Then, when areas of detection are well defined, I can play with the threshold on heatmap detection to be sure to detect only cars.

6 Results

Finally, with this method, the results were pretty good on images:

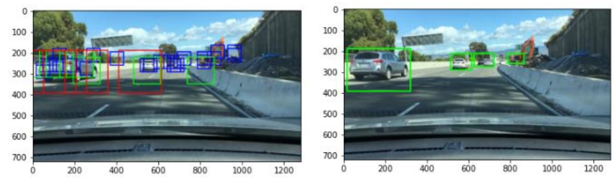

Figure 4: Sliding windows and bounding boxes

I obtain a Sørensen-Dice coefficient of 40%, which is a very good result.

7 Conclusion

In conclusion, the implementation of a car detector using HOG features and SVM classifier has proven to be highly effective. Outside Deep Learning, this algorithm seems to be the most effective method [3]. The results achieved were impressive, with the detector able to accurately identify cars with a high level of precision. Furthermore, the simplicity of the code and the ease of implementation make this approach an attractive option for real-world applications. Overall, the success of this project highlights the importance of selecting appropriate feature extraction techniques and classification algorithms to improve the performance of computer vision systems.

8 References

[1] https://www.kaggle.com/datasets/brsdincer/vehicle-detection-image-set

[2] Nikola Tomikj, Andrea Kulakov. Vehicle Detection with HOG and Linear SVM. Journal of Emerging Computer Technologies. 30-06-2021. 4.

[3] Hilton Bristow, Simon Lucey. Why do linear SVMs trained on HOG features perform so well?. 10-06-2014. 8.