Automatic Detection of Cars in Real Roads using Haar-like Features

M. Oliveira, V. Santos

Department of Mechanical Engineering, University of Aveiro, 3810 Aveiro, Portugal.
{mriem, vitor}@ua.pt

Abstract: This paper describes a computer vision based system designed for the detection of cars in real world environments. The system uses the Haar-like features method firstly introduced by Viola and Jones which is known for its fast processing and good detection rates. The process requires representative data sets to be used for training and validation including positive (presence of objects to detect) and negative (absence of objects to detect) image samples. Therefore, several example images of cars were hand labeled for training and performance calculation purposes. Preliminary results show that the method can be very effective to detect cars at fast rates and show generalization capabilities. Despite some occasional false detections, because this method is quite fast, it can act as a primordial filter of promising regions of the image, where more effective yet time demanding tests can later be employed.

Keywords: Haar Features, Computer Vision, Automatic Detection of Cars, Transportation systems,

1 INTRODUCTION

Automatic navigation in real roads is an old aspiration of road drivers, and of the automotive industry in general, because of the large importance it can one day reach in what concerns security. Indeed, annually, all over the world, many casualties occur on the road due to accidents and are often caused by driver distraction or lack of responsiveness in demanding driving conditions (traffic, weather, individual focusing on the driving tasks, etc.).

The future of automatic navigation in roads will necessarily require advanced and robust perception of the road and traffic entourage, which can be very complex due to the huge variety of subjects and conditions (roads, vehicles, illumination and weather, etc.). Roads and vehicles are among the most relevant subjects in that framework therefore, they represent a must when starting to develop systems for automatic navigation on the road detection.

While the authors have been questing for autonomous navigation in road-like tracks in a parallel research activity [1][2] this paper focuses specifically on one method for automatic car detection. The technique uses Haar-like features, and cascade classifiers are "trained" to match cars in real roads. A paper regarding the detection of a single object using Haar features was already published by the authors [3]. The paper introduces briefly the Haar-like features concept, then describes the cascades used specially for cars, and before the conclusion presents extensive results on untrained road images under varied circumstances.

2 HAAR-LIKE FEATURES

Haar-like features were proposed by Viola and Jones [4] as an alternative method for face detection. The general idea was to describe an object as a cascade of simple feature classifiers organized into several stages. This is a very fast method, performing face detection as effectively as any other methods. As stated in [4], in the CMU+MIT reference test set, the method performed 15 times faster than the Baluja-Kanade detector and about 600 times faster than the Schneiderman-Kanade detector.

The classification of images is based on the value of simple basic features. Features are used instead of simple raw pixel values generally because they can act to encode *ad-hoc* domain knowledge but also, in this particular case, because they are much faster to process.

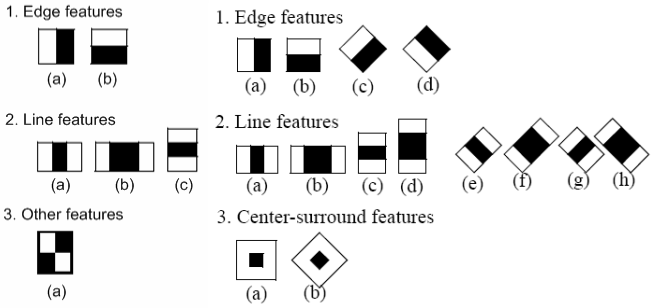

Figure 1. Basic set of Haar features used by [4] (left), and extended set applied by [5] (right). Taken from [5].

Later on, Lienhart and Maydt proposed to extend the pool of basic features by utilizing also 45º rotated features thus "significantly enhancing the expressional power of the

learning system and consequently improving the performance of the object detection system" [5]. The features that were proposed by Viola and Jones (the basic set) and latter by Lienhart and Maydt (extended set) are shown in Figure 1. It is important to emphasize that the features of Figure 1 are mere prototypes. They are scaled independently in horizontal and vertical directions in order to get an over complete set of features (the 24x24 window proposed in [4] the amount of possible features is around 180000). The result of the application of each feature to a particular image region is given by the sum of the pixels that lie within the black rectangles of the feature subtracted by the sum of the ones overlapping the white rectangles. The rectangles are defined by their top left coordinates x, y, their width w and height h. The sum of the pixels that lie within the rectangle r_i is represented by $\mathrm{RecSum}(r_i)$.

$$feature_1 = \sum_{i=1}^{N} W_i \times \mathrm{RecSum}(r_i) = \\ = \sum_{i=1}^{N} W_i \times \mathrm{RecSum}(x, y, w, h) \quad (1)$$

The values of N, W_i and of r_i are arbitrarily chosen. In the case of [5], it has been defined that $N = 2$ and that black rectangles (r_0) have negative weight W_i and white (r_1) have positive weights. Furthermore, the relationship between weights is given by the difference of area occupied by the black and white rectangles.

$$-W_0 \cdot Area(r_0) = W_1 \cdot Area(r_1) \quad (2)$$

Assuming $W_0 = -1$, one can obtain:

$$W_1 = \frac{Area(r_0)}{Area(r_i)} \quad (3)$$

Consequently, for example for feature (2a) of Figure 1, with a height $h = 2$ and width $w = 6$ the outcome of the feature application to a rectangular region positioned at x, y would be:

$$feature_{2a} = -1 \cdot \mathrm{RecSum}(x, y, 6, 2) + \\ + \frac{6 \times 2}{2 \times 2} \cdot \mathrm{RecSum}(x + 2, y, 2, 2) \quad (4)$$

In order to compute the value of each feature very rapidly, an intermediate image representation is calculated. This representation is called integral image or Summed Area Table (SAT). The value of the integral image at coordinates (x, y), is given by the sum of all the pixels in the image that are above and to the left of (x, y):

$$SAT(x, y) = \sum_{x' \le x, y' \le y} I(x', y') \quad (5)$$

Where $I(x', y')$ is the value of the image's pixel at coordinates (x, y). The value of any $\mathrm{RecSum}(x, y, w, h)$ can be obtained by simply four lookups at the SAT.

$$\mathrm{RecSum}(x, y, w, h) = \\ = SAT(x + w, y + h) - SAT(x + w, y) - \\ - SAT(x, y + h) + SAT(x, y) \quad (6)$$

This procedure is shown on Figure 2.

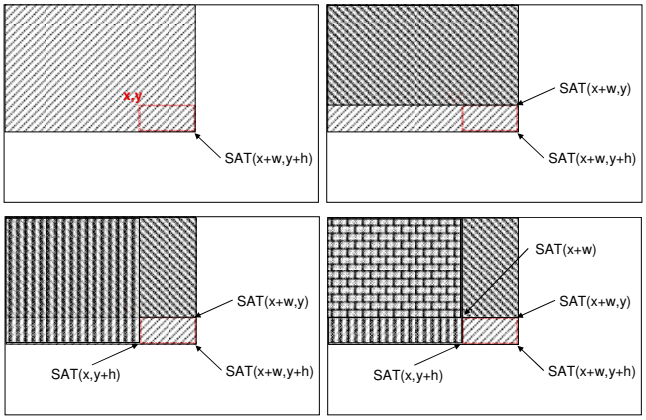

Figure 2. Fast $\mathrm{RecSum}(r_i)$ calculation.

Viola and Jones [4] set up a framework to combine several features into a cascade, i.e. a sequence of tests on the image or on particular regions of interest, organized into several stages, each based on the results of one or more different Haar features. For an object to be recognized, it must pass through all of the stages of the cascade. The cascade is built by supplying a set of positive and negative examples to the training algorithm. The used algorithm is called Adaboost, known for its high performance in what concerns generalization speed [4]. At each stage of the cascade, the machine learning algorithm selects the feature or a combination of features that best separate negative from positive examples, by tuning the threshold classification function. There is a trade off relationship between the number of stages in a cascade and features in each stage and the amount of time it takes to process the cascade. Viola and Jones define, for each stage, a target for the minimum reduction in false positives and a maximum decrease in detection. The mentioned rates are obtained by using a validation set made up of the positive and negative examples. In order to improve the time performance of the algorithm, the same authors have also presented the notion of attentional cascade. The idea consists of using the first stages of the cascade to effectively discard most of the regions of the image that have no objects. This is done by adjusting the classifier's threshold so that the false negative is close to zero. By discarding many candidate regions early in the cascade, Viola and Jones significantly improve the method's performance. In fact, it makes a lot of sense that the detection system is able to quickly discard obvious negative regions of an image using valuable time to better test much

more promising regions by submitting them to higher level stages of the cascade that yield more complex features.

It has already been said that Haar-like features were especially applied to perform face detection. However, the framework is all-purpose. Some other approaches have successfully used it for pedestrian detection [6].

3 A Generalized Cascade for Car Detection

The next attempt was to train a generalized cascade for the detection of car's rears, i.e. to train a Haar cascade that would detect not a particular car model, but an indiscriminate car's rear detector. Haar features are first trained to obtain a representation to be used latter for real time object detection. For this purpose several image collections were acquired. They will be described in the subsequent chapters.

3.1 Training Datasets Description

For training purposes, two image datasets were borrowed from the internet and a third was made by the authors. This chapter will describe in detail each set, indicating the number of images per set, their properties and locations where they were taken. Table 1 sums up the training sets information. Training datasets will, henceforth, be named as TDS followed by their respective number.

Table 1. Training datasets description.

Name	Nº Img	Resolution	Location	Authors
TDS 1	1556	variable	California	unknown
TDS 2	126	896x592	California	Weber
TDS 3	1004	752x512	Portugal	Oliveira, Santos

California Institute of Technology dataset is composed of 1156 images in *png* format, though many are very similar (Figure 3). Image resolution is variable. This dataset is used for training and will henceforth be named training dataset 1 (TDS1).

Figure 3. Samples from of California Institute of Technology dataset.

Markus Weber's dataset is not as broad, bearing only 126 images. The resolution is 896x592 pixels, *jpg* format and the images were taken in the California Institute of Technology's parking lots. Some examples are on Figure 4. This will be named training dataset 2 (TDS2).

Figure 4. Car dataset taken by Markus Weber, California Institute of Technology.

A third dataset was made by the authors during a car travel in Portugal (from Algarve to Aveiro) and nearly 2 hours of footage was captured. Resolution was 752 x 512 pixels. Over 1000 images were extracted from the film. Positive examples were separated and cars were also hand labeled (Figure 5). No rescaling was performed. This will be referred to as training dataset 3 (TDS3).

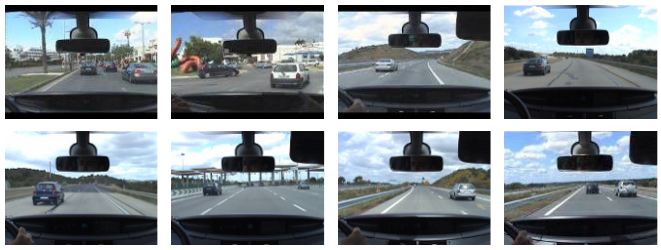

Figure 5. Authors' own car dataset from Portuguese roads.

Some of the images were taken during adverse weather conditions, such as rain. Some examples are present on Figure 6. These images are also included in TDS3.

Figure 6. Authors' own car dataset with poor weather conditions.

3.2 Performance Datasets Description

For the purpose of testing, three separate datasets are used. The first dataset was built by Brad Philip and Paul Updike (Figure 7).

Figure 7. Car dataset taken by Brad Philip and Paul Updike, California Institute of Technology.

It was taken on the freeways of southern California. It is composed of 530 images in *jpeg* format. Resolution is constant at 320x240 pixels. Images are quite similar to TDS1 but are not included in it. This test dataset will be employed to measure the performance of the cascades and will be

mentioned as performance dataset 1 (PDS1).

Performance dataset 2 (PDS2) is taken from the footage that provided images for TDS3. The images are not the same although they are similar. PDS2 consists of 105 images, 756x512 pixels of resolution, saved in png format. No demanding weather conditions, city environment or gas stations images were included. The idea was to use a simplified version of the footage. Finally, performance dataset 3 (PDS3) is an extension of PDS2 obtained by including all kinds of images: poor weather, city, gas stations, bridges etc. (Figure 8).

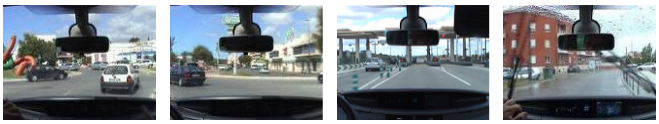

Figure 8. Complex images in PDS3.

PDS3 is a much harder set. It consists of 232 images with the same resolution and format as the ones of PDS2.

Table 2. Performance datasets description.

Name	Nº Img	Resolution	Location	Authors
PDS 1	530	320x240	California	Philip, Updike
PDS 2	105	752x512	Portugal	Oliveira, Santos
PDS 3	232	752x512	Portugal	Oliveira, Santos

3.3 Cascades Description

Having ensured a wide variety of examples, hand labeling was performed over all images both in training and performance sets. A semi-automatic hand labeling application was developed to ease the process by enabling fast mouse selection. It also generates a text file were the ROI or ROIs (i.e. the regions were the car or cars can be found) is/are defined for every image. Intel Open Source Computer Vision Library (Opencv) [7] provides a tool that creates samples by clipping the defined ROIs from TDS images, converting them to grayscale, rescaling them to window size, and inserting them into a random background image. The background image pool, or negative set, has not yet been described. A negative set consists of a set of images where no objects (cars) exist. They haven't been mentioned since they are impaired with their respective TDS, i.e., every TDS also has a set of negative examples, usually road images where no cars are present.

Table 3. Cascades Description.

Name	Win Size	Train. Set(s)	T. Samples (pos/neg)	Nº Stages	Features Set
C1	30x20	1 + 2	unknown	20	BASIC
C2	60x40	1 + 2	1282 / 754	20	BASIC
C3	30x20	3	unknown	20	BASIC
C4	30x20	3	unknown	20	ALL
C5	60x40	3	unknown	20	BASIC
C6	30x20	1 + 2 + 3	1556 / 915	20	BASIC
C7	30x20	1 + 2 + 3	1556 / 915	20	ALL
C8	20x12	1 + 2 + 3	1556 / 915	30	ALL

Table 3 summarizes the setup used for several cascades trained using different combinations of TDS, number of stages, features pool, i.e., BASIC for Viola Jones features collection and ALL meaning Lienhart and Maydt extended set as well as the number of positive and negative samples generated after the dataset (T. Samples). Also, several window sizes were attempted. Cascades will henceforth be named by C followed by their respective number.

4 Performance Tests and Results

Opencv [7] provides a tool for cascade performance testing. The tool applies the cascade to all test images and compares the algorithm's outcome to the report generated by hand labeling. Hit (HR) and false detection (or false alarm) (FDR) rates are generated based on this comparison. In order to assume a given detection as the one described in the report, some tolerances are assumed. Tolerances are related to the disparities in position and size from the current detection and the one manually generated for comparison For every detection made, a search in the corresponding PDS is executed to see if the detection is true or false. In order to allow for easy performance comparison, the tolerances employed are the default values of the mentioned tool. PDS1 was tested with several cascades and several scaling factors scaling factor, sf , which is a Haar detection parameter that indicates how much the reference window should be scaled up. HRs are quite good (some above 95%) though FDRs are quite high (Table 4).

Table 4. Performance results for PDS1.

Name	sf	Hits	Missed	False Detect.	HR	FDR
C1	1.05	508	18	400	0,966	0,760
	1.9	370	156	263	0,703	0,500
C2	1.05	501	25	444	0,952	0,844
	1.5	501	25	193	0,952	0,367
	2.9	311	215	500	0,591	0,951
C3	1.05	440	86	4840	0,837	9,202
	1.9	436	90	1529	0,829	2,907
C5	1.05	449	77	2676	0,854	5,087
	1,9	495	31	953	0,941	1,812
	2,9	397	129	874	0,755	1,662
C6	1.05	442	84	5799	0,840	11,025
C7	1.05	429	97	3385	0,816	6,435
	1,9	396	130	925	0,753	1,759
	2,9	193	333	868	0,367	1,650

There is a trade-off relationship between HR and FDR. Detecting a given feature with a HR of 100%, would obviously raise the FDR. The results here shown present the HR and FDRs of the complete cascades, i.e. the cascades with all the stages included. The tables show the results for the maximum achieved HR, and the FDR associated with them. It is important to bear in mind that, though the FDRs

presented in Table 4, Table 5 and Table 6 may seem quite high, the charts on Figure 9, Figure 11 and Figure 13 provide a much better view of the HR versus the FDR relationship. A close analysis of those figures clearly shows that a small loss in the HR would imply a large reduction of the FDR. Also, some techniques that may considerably reduce the FDRs are discussed ahead.

The cascades that best perform would be $C1_{sf=1.05}$ and $C2_{sf=1.5}$. The ROC curves for both are presented at Figure 9.

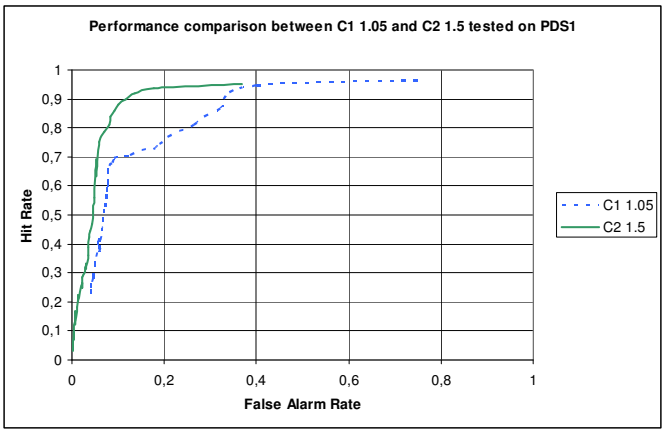

Figure 9. ROC of the best performing cascades on PDS1.

Figure 9 shows that cascade $C2_{sf=1.05}$ performs better than $C1_{sf=1.05}$. Cascade $C2_{sf=1.5}$ can achieve the same HR as $C1_{sf=1.05}$ at a lower cost, i.e., lower FDR. Analyzing Figure 9 one could assume the optimum point to be $HR \simeq 0.92$ and $FDR \simeq 0.18$. This would imply that 92% of all cars were detected yielding only 18 false detections per every 100 truthful ones. Some examples of $C2_{sf=1.5}$ detections can be seen on Figure 10.

Figure 10. Examples of detections made by $C2_{sf=1.5}$ on PDS1.

Regarding PDS2, fewer tests were executed. Table 5 clearly shows a much higher FDR's average score.

Table 5. Performance results for PDS2.

Name	sf	Hits	Missed	False Detect.	HR	FDR
C4	1.05	116	28	2150	0,806	14,931
C5	1.05	79	65	1213	0,549	8,424
C6	1.05	84	60	1081	0,583	7,507

While $C4_{sf=1.05}$ yields the best HR, it also has a FDR of 14, i.e. for every detection that should be made, 14 false detections occur. This number may appear high if the cascade is used for actual detection but may loose relevance if the cascade is to be used as a simple attention mechanism or if further validation tests are to be implemented.

Figure 11. ROC of the cascades that best performed on PDS2.

On Figure 11, the optimum point of cascade $C4_{sf=1.05}$ presents values of $HR \simeq 0.78$ and $FDR \simeq 0.75$. Tests with PDS2 were not entirely satisfactory in what concerns FDRs. However, this is a very difficult set and some additional procedures could have been implemented to ease the FDR and also, in some cases, improve the HR.

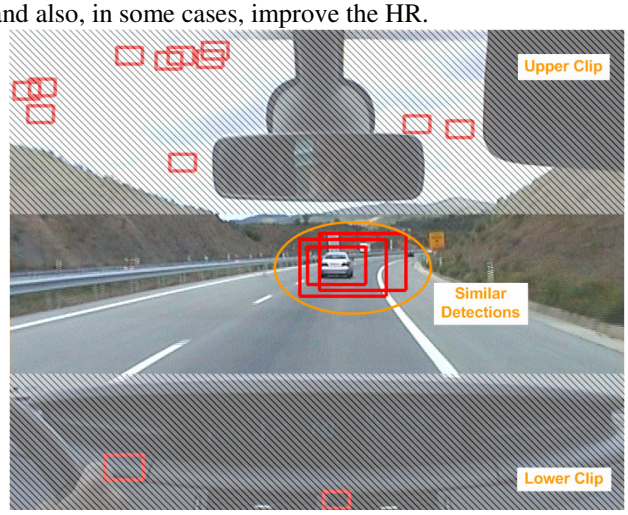

Figure 12. PDS2's apparent problems.

First of all, the images from PDS3 could be clipped without loss of reliable extrapolation of the algorithm's performance (Figure 12). The upper and the lower parts of these images contain no information on the road (sky/rear mirror and car interior panel). This clipping operation would lower considerably the FDR since many of these false detections are in these areas of the images. Also, many of the detections are very close to each other. An algorithm for merging overlapping detections (or a fine tune of the performance calculation tolerances mentioned at the beginning of this chapter) could be easily implemented thus decreasing even more the FDR. In the case of Figure 12, one would go from a situation with 15 false detections to none, if these procedures were implemented, which would dramatically lower the FDR. Taking the previous considerations into account, it seemed interesting to test

some cascades on PDS3, even knowing that it is even more demanding than PDS2. The results are outlined on Table 6

Table 6. Performance results for PDS3.

Name	sf	Hits	Missed	False Detect.	HR	FDR
C1	1.5	78	249	156	0,239	0,477
C2	1.5	89	238	612	0,272	1,872
C3	1.5	269	58	1510	0,823	4,618
C4	1,05	256	71	4837	0,783	14,792
	1.5	204	123	2028	0,624	6,202
C5	1.5	158	169	1252	0,483	3,829
	1.05	79	65	1213	0,549	8,424
C6	1.5	158	169	1457	0,483	4,456
C7	1.5	115	212	1649	0,352	5,043
C8	1.05	295	32	4119	0,902	12,596
	1,9	30	297	43	0,092	0,131
	2,9	189	138	837	0,578	2,560

In this particularly difficult dataset, most of the cascades present a low HR. However, cascades $C3_{sf=1.5}$, $C4_{sf=1.05}$ and particularly $C8_{sf=1.05}$ have acceptable HRs. Of course that the FDRs are considerable, but there is the conviction that these rates can be substantially reduced by means of the already mentioned clipping and merging techniques.

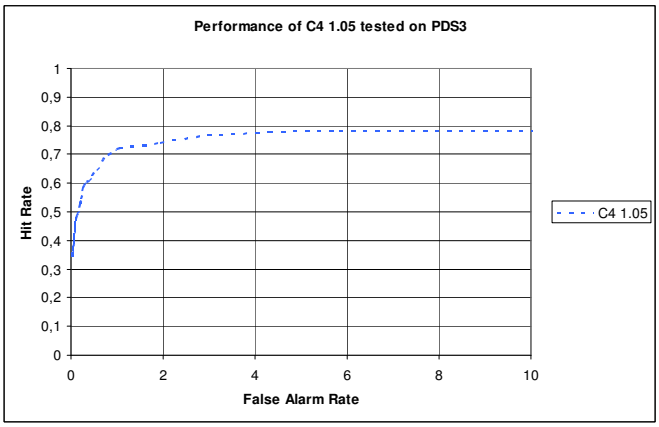

Figure 13. ROC curve of $C4_{sf=1.05}$ tested on PDS3.

Performance data was not extracted from $C3_{sf=1.5}$ neither from $C8_{sf=1.05}$ and so Figure 13 presents only the results of $C4_{sf=1.05}$. The optimum point of detection performance for cascade $C4_{sf=1.05}$ would, nonetheless, yield acceptable $HR \simeq 0.72$ and $FDR \simeq 1$. Bearing in mind that FDRs could be overstated and that PDS3 is a set of high complexity, including images in the rain, city and other tricky obstacles, the HR of $C8_{sf=1.05}$ is quite acceptable (Figure 14).

5 Conclusions And Future Work

This paper presented a method based on Haar-like features designed for the detection of cars in real roads. Three TDS were used for the training of the cascades. Also, three different PDS were employed for performance testing. The best achieved results show $[\, HR \simeq 0.92 \,;\, FDR \simeq 0.18 \,]$; $[\, HR \simeq 0.78 \,;\, FDR \simeq 0.75 \,]$ and $[\, HR \simeq 0.72 \,;\, FDR \simeq 1 \,]$ respectively for PDS 1, 2 and 3.

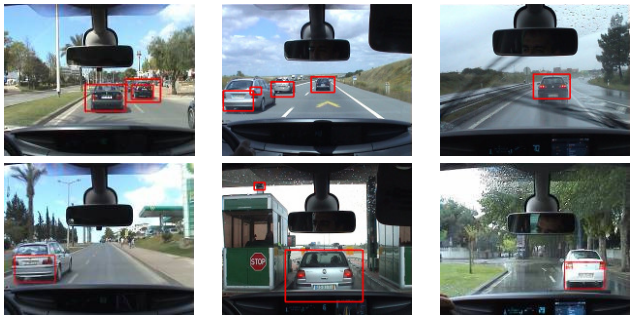

Figure 14.Some detections of PDS3.

Some methods for decreasing the FDRs were suggested and will be further explored in the future. Most of the cascades were tested against all PDSs, which may provide relevant information regarding the influence of variables such as window size, training sample size and variability, usage of rotated Haar features and others on the cascade performance.

In the case of PDS1, $C2_{sf=1.5}$ performed better than $C1_{sf=1.05}$, which seems to corroborate the idea that a larger detection window may best describe an object (review Table 3).

Regarding PDS2, $C4_{sf=1.05}$ is more efficient than both $C5_{sf=1.05}$ and $C6_{sf=1.05}$. The increase in performance may be due to the usage of both simple and rotated Haar features. The processing of the cascades is quite fast, which enables the future implementation of this method in a real time system.

6 References

[1] R. Cancela, M. Neta, M. Oliveira, V. Santos, 2006. *ATLAS III: Um Robô com Visão Orientado para Provas em Condução Autónoma*, Robótica, nº 62, 2006, p. 8 (ISSN: 0874-9019).

[2] M. Oliveira, V. Santos , *A Vision-based Solution for the Navigation of a Mobile Robot in a Road-like Environment*, Robótica, nº69, 2007 p. 8 (ISSN: 0874-9019).

[3] M. Oliveira, V. Santos. *Combining View-based Object Recognition with Template Matching for the Identification and Tracking of Fully Dynamic Targets,* 7th Conference on Mobile Robots and Competitions, Festival Nacional de Robótica 2007. Paderne, Algarve. 27/04/2007

[4] P. Viola, M. Jones 2001. *Rapid Object Detection using a Boosted Cascade of Simple Features*, Conference on Computer Vision and Pattern Recognition CVPR, Hawaii, December 9-14, 2001.

[5] R. Lienhart and J. Maydt. *An Extended Set of Haar-like Features for Rapid Object Detection*. IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep. 2002.

[6] G. Monteiro, P. Peixoto, U. Nunes, 2006. *Vision-based Pedestrian Detection using Haar-like Features*. Encontro Científico, Festival Nacional de Robótica 2006.

[7] Opencv, Intel Open Source Computer Vision, found at *http://www.intel.com/technology/computing/Opencv/* on January 2007.