Vehicle Detection
using Images from Traffic Security Camera

Lamia Iftekhar

Final Report of Course Project
CS174
May 30, 2012

1 The Task

This project is an application of supervised learning algorithms. Our objective is to detect vehicles in still images obtained from an intersection traffic camera.

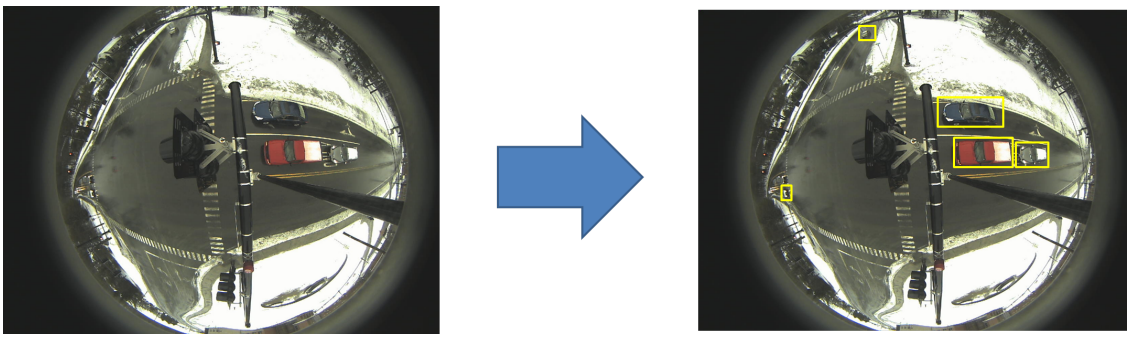


Figure 1: A conceptual illustration of the project objective

2 The Original Dataset

The raw dataset used for this project is obtained from the Hanover Police Department. It is a collection of still images taken from a single traffic camera mounted on the Lyme/ North Park intersection near Dartmouth Medical School. This database contains images taken during late January 2012. For this project, we restrict ourselves to only daytime images.

3 The Approach

We used the Viola-Jones face detector method [4] to detect vehicles from the image. Although this method was originally designed to detect faces, the concept presented in the paper can be applied to detecting objects of other kinds provided suitable training sets are used. Several papers have demonstrated so in the context of vehicle detection [2],[1],[3].

3.1 Brief Description of the Viola Jones Detector

The Viola-Jones framework has three important characteristics:
- its feature selection process based on integral images and Haar-like features: sums of image pixels within rectangular areas
- its learning method based on training weak learners and blending their outputs to make a strong classifier, reweighting examples at each training step (Adaboost)
- its 'attentional cascade' structure which builds on the premise that when scanning an image for faces, the detection speed can be increased if negative regions are quickly discarded using initial classifiers and more complex classifiers only focus on the more promising regions of the image. (see Figure 3).

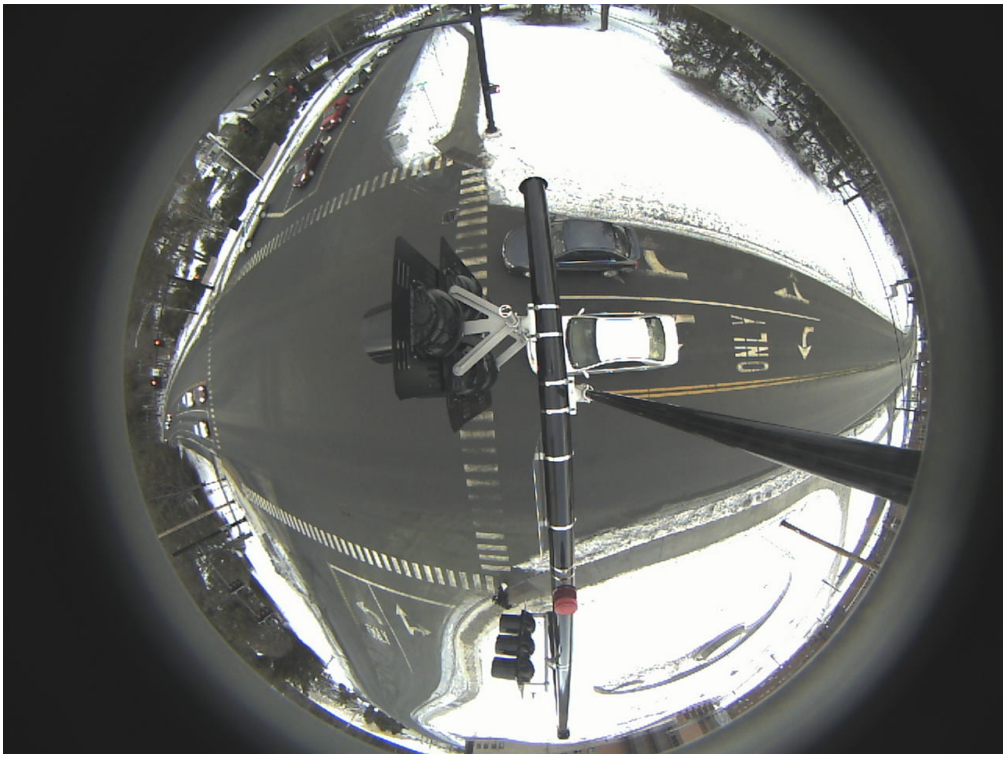


Figure 2: A sample image from the dataset

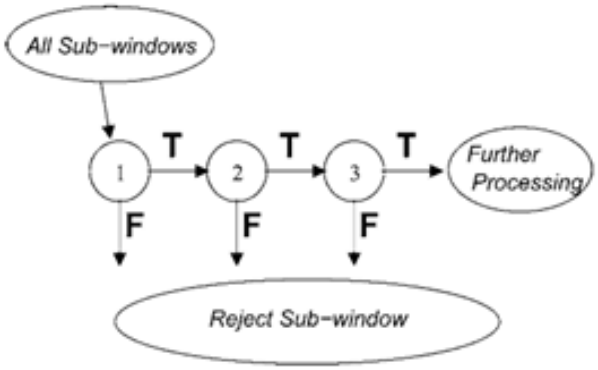


Figure 3: Depiction of detection cascade (figure source: [4])

The Viola-Jones detector uses Adaboost for both feature selection and learning. The early features chosen by Adaboost are meaningful. For example, for face detection, Haar-like objects such as those shown in Figure 4 are chosen and they may represent the contrast between the eyes and nose or the eyes and cheeks. For vehicle detection similar features may represent the dark region underneath a car or the two tires (Figure 5) .

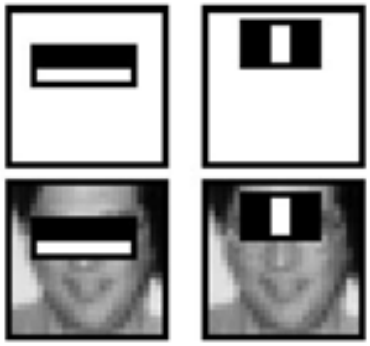


Figure 4: Haar-like features used for face detection (figure source: [4])

Figure 5: Haar-like features used for car detection (figure source: [4])

3.2 Project Strategy

One of the biggest challenge of our project is that, unlike the datasets used in the previous works mentioned above, we do not have pictures of vehicles from the same angle, i.e., background or foreground. On the other hand, our images includes vehicles from various angles due to the location of the camera and the fish-eye view. Appearance of vehicles vary greatly depending on where the vehicles are located in relation to the intersection. For example, the vehicles on the right of the intersection are seen from the top mostly, whereas the vehicles coming down from the top road are seen more from the side. Also the size of the vehicles are highly distorted depending on their position in the image. Hence, we train separate classifiers for each of the four roads coming into the intersection. Approximate location of the 'focus area' for each of the four classifier are given in Figure 6.

Moreover, we do not implement the cascaded detector, because the main advantage of a cascaded detector over a monolithic one is its speed, but both have the same accuracy. Figure 7 is a graph reproduced from [4] to illustrate this point.

4 Project Work

In this section we describes what has been done for the project and how we did it.

4.1 Data Processing

We began by building training database for each of the four classifiers. This had to be done manually, as standard available databases consisted of the cropped images of vehicles from a fixed angle and that would not help us build classifiers that can detect vehicles from fish-eye images as such as ours.

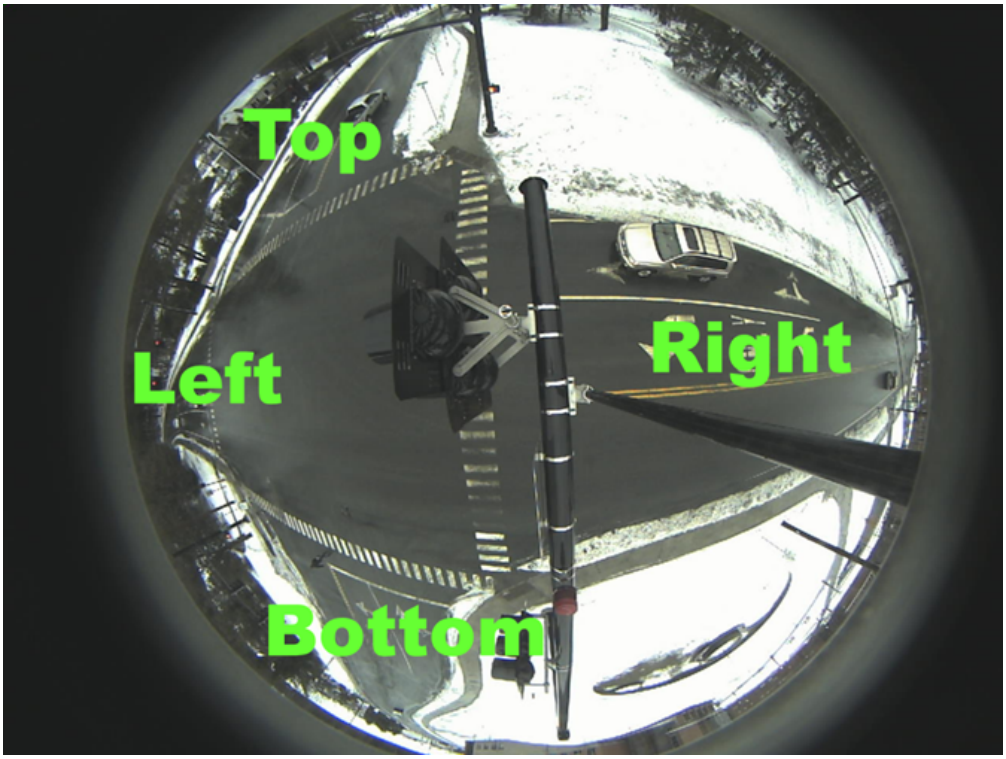


Figure 6: Approximate regions of focus for the four different classifers

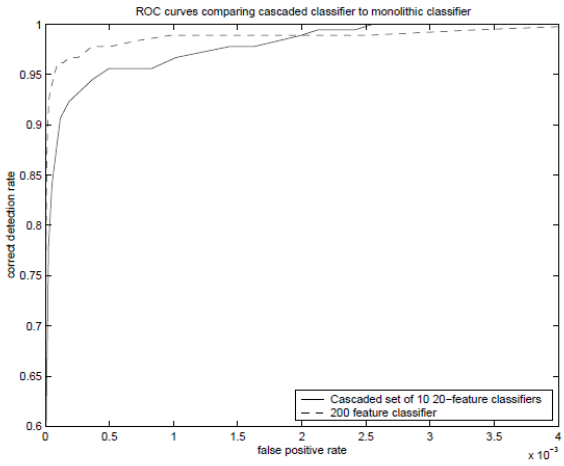


Figure 7: ROC curves comparing a 200-feature classifier with a cascaded classifier containing ten 20-feature classifiers. Accuracy is not significantly different, but the speed of the cascaded classifier is almost 10 times faster.

Figure 7: Performance of cascaded detector vs. a monolithic detector

For each of the four focus areas in the traffic camera images, we built a collection of positive and negative samples by cropping out regions with vehicles and no vehicles respectively. We set aside some of

the cropped images as validation dataset.

Each cropped color image was then turned to grayscale image, histogram equalized and finally scaled and normalized to the same size. The quantitative details are provided in the table in Figure 9.

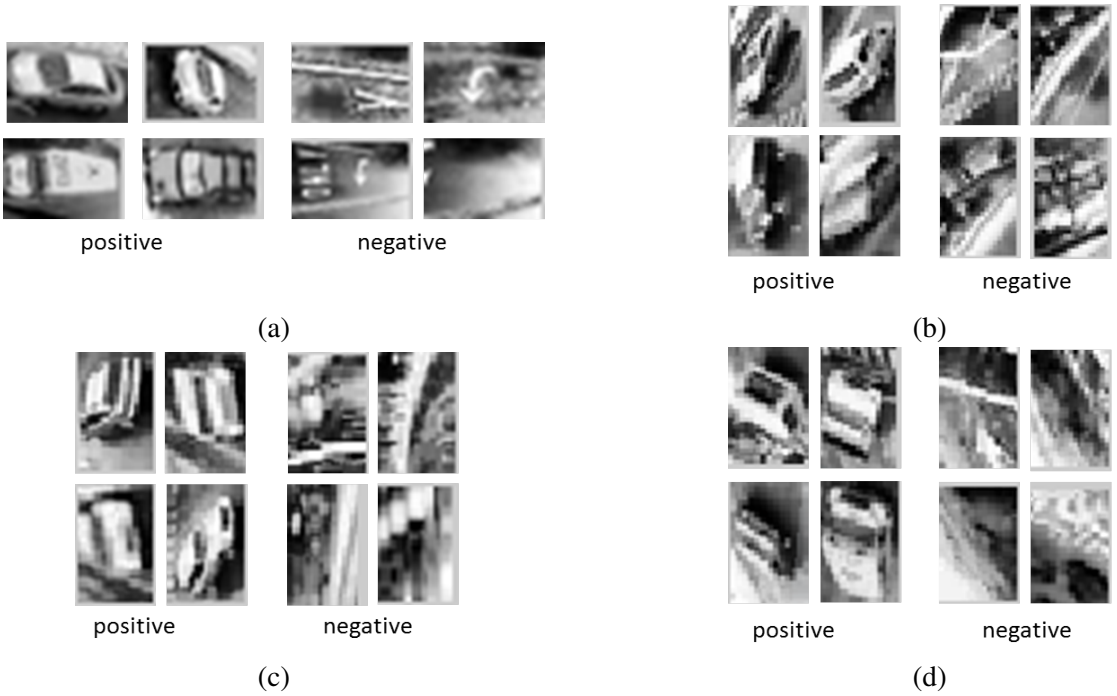


Figure 8: Examples of pre-processed training samples from each focus area. (a)right, (b)top, (c)left, (d)bottom

Focus Area	Number of Training Samples	Number of Validation Samples	Resizing (in pixels)
Right	200 (positive : 150 ,negative: 50)	100 (p:50, n:50)	20 by 30
Top	207 (p: 155, n: 52)	102 (p:52, n:50)	30 by 20
Left	100 (p: 50, n: 50)	50 (p:30, n:20)	30 by 20
Bottom	400 (p: 202, n: 198)	100 (p:51, n:49)	30 by 20

Figure 9: Training set details

4.2 Training

We train each of the four classifiers in exactly the same way described in the Viola-Jones paper, for the non-cascade detector (their algorithm is reproduced in Figure 10).

Here T is the desired number of weak classifier we want to have in our final strong classifier. We chose T=10.

The weak classifiers used are single node decision trees, aka decision stumps. Each weak learner depends on a single feature. For each feature, the weak learner chooses the optimal threshold classication function, such that the minimum number of examples are misclassied. A weak classier $(h(x, f, p, \theta))$ consists of a feature (f), a threshold (θ) and a polarity (p) indicating the direction of the inequality:

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta \\ 0 & \text{otherwise} \end{cases}$$

where x is a 20 by 30 (or 30 by 20) image.

Table 1. The boosting algorithm for learning a query online. T hypotheses are constructed each using a single feature. The final hypothesis is a weighted linear combination of the T hypotheses where the weights are inversely proportional to the training errors.

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights, $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
  2. Select the best weak classifier with respect to the weighted error
     $$\epsilon_t = \min_{f,p,\theta} \sum_i w_i \, |h(x_i, f, p, \theta) - y_i| \, .$$
     See Section 3.1 for a discussion of an efficient implementation.
  3. Define $h_t(x) = h(x, f_t, p_t, \theta_t)$ where f_t, p_t, and θ_t are the minimizers of ϵ_t.
  4. Update the weights:
     $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
     where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:

$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

Figure 10: Viola-Jones training algorithm (non-cascaded version) as presented in the [4]

As mentioned before, the features used are Haar-like features. These are simply the difference in the sum of the pixels which lie within the white rectangles and the sum of pixels in the dark rectangles. Integral images are used for ease of computing these differences. In the Viola-Jones paper, four Haar-like feature

types are used (1,2,4,5 from Figure 11). We added feature type 4 (horizontal three-rectangle features)as we believed that if feature 3 is helpful in detecting vehicles in any of the region, feature 4 would surely help in detecting vehicles, at least in the regions orthogonal to the former.

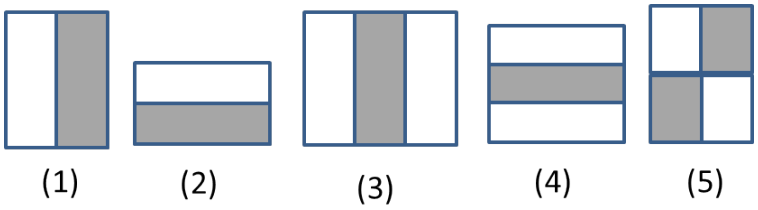


Figure 11: Types/shapes of features used

For each training image, one feature type can generate numerous actual features. For example, for a 20 by 30 image, for feature type 1, the first feature would like in Fig 12(a), then we shift and obtain the second feature as shown in (b)and continue with all possible translation with this fixed size. We then obtain more features by increasing the scale as in (c) and continuing translation as before. For a 20 by 30 image, just using feature type 1 results in having to evaluate more than thirty-three thousand features!

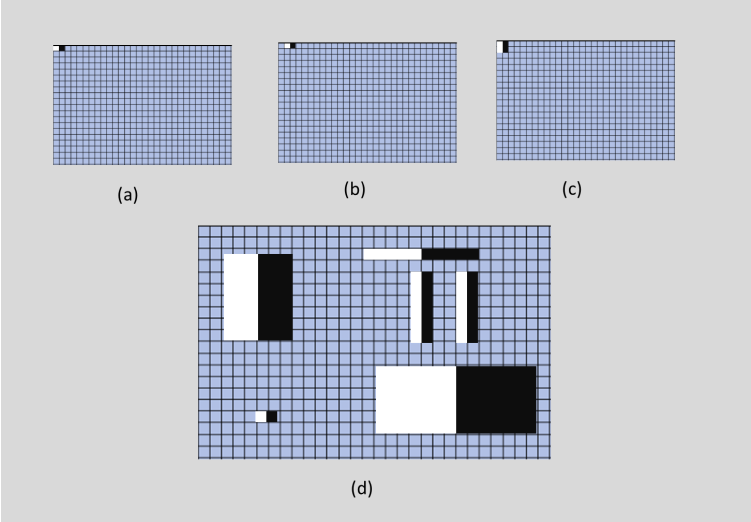


Figure 12: Explanation of Haar-features in an image. (d) shows some of the different features possible with only feature type 1

4.3 Testing

First we test, each of the classifiers learned on their respective validation set (see table in Figure 9 for the number of cropped images used for this purpose).

We then implement a MATLAB program that can take a full image (an image of the whole intersection from the original database) and use sliding windows to apply the appropriate classifier for each of the four focus area, and output the original image with bounding boxes around regions it classifies as vehicles. The sliding windows vary in size depending on the focus area, but for a given focus area, they have a fixed scale.

5 Results

The missclassification rate for each classifier on its own validation set is given by the graph below.

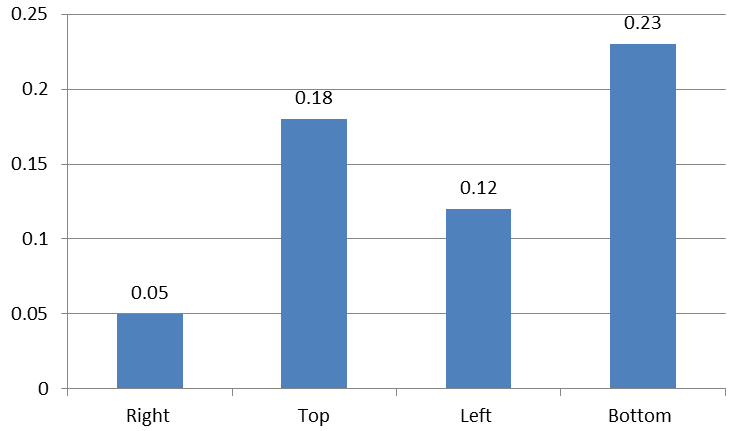


Figure 13: Error rate of each strong classifier

Within the missclassified samples by each classifier, the ratio of false positives can be deduced from the graph in Figure 14.

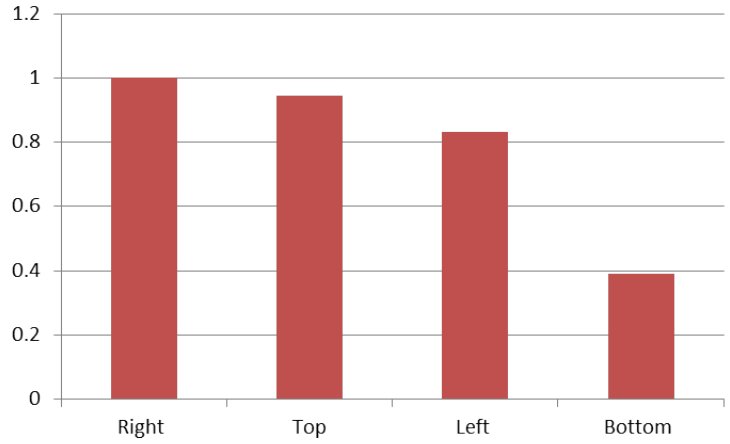


Figure 14: False positive amongst the missclassified samples in each strong classifier

Figure 15 provides an example output of the sliding window test program implementing the full detector consisting of the four classifiers. The high rate of false positives is noticeable.

6 Discussion

Viola-Jones detection method is a very slow learner. For a proper training dataset of thousand images and building a classifier with T=200, it takes about a week to learn. This project uses a much smaller dataset and T=10 only, yet it takes about four hours to train one classifer. Hence, due to limited time , not much

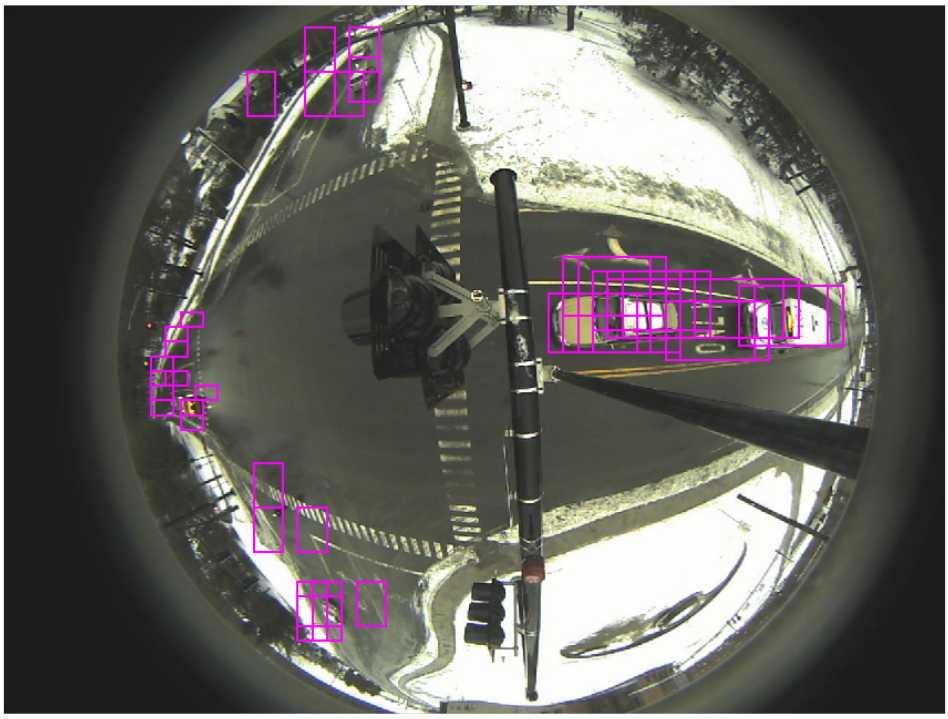


Figure 15: An example output of the detector utilizing location-dependent classifiers

analysis could be in the context of training multiple classifiers with different features etc, and comparing them. The following discussion is based on training each of the four classifiers one time only, where each trained classifier used five feature types and consisted of T=10 weak classifiers.

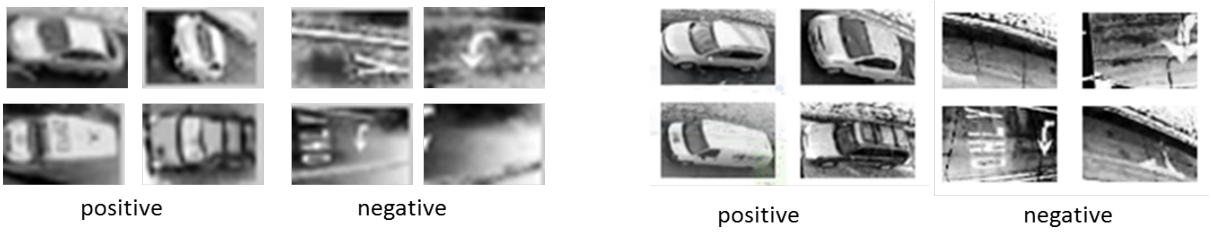
Based on the training errors in Figure 13, the 'Right' classifier provided the best result. This is reasonable as the vehicles in the right focus area have the most uniform appearance, which is the top view of the vehicles. In fact, the initial weak classifiers chosen by Adaboost for this focus area are of type 1, which possibly captures the relatively darker region of the windshield of a vehicle when viewed from the top. Classifiers for the top and bottom focus areas have lower rates of success, as the vehicles in the area are less uniform in appearance and view of these vehicles changes drastically with the slightest difference in location. Figure 16

Figure 16: Variety in the angle/appearance of vehicles in the bottom road (taken from training set)

From the outputs of the sliding window program, one can observe that even though our detector detects all vehicles correctly, the number of false positives is very high. Also, there are multiple detections around one vehicle, but this phenomenon has been mentioned in the Viola-Jones paper to be something expected, as the final detector is insensitive to small changes in translation. But this could be easily mended with some postprocessing.

6.1 Future Work

Suggestion for future work, first and foremost includes, utilizing a much larger training set. We should also train classifiers with a larger number of weak classifiers than just T=10. Also, as the appearances of the vehicles, even within a given focus area, vary so greatly, the number of features required to detect vehicles are high too. Hence, 20 by 30 (or 30 by 20) size of training images might be two small. A database of training images 40 by 60 pixels (or 60 by 40 pixels) in size may train classifiers better.



(a) (b)

Figure 17: Comparision of some 20x30 training samples (a) vs. some 40x60 training samples(b). 20x30 training images might not have enough information to train an efficient classifier

We could also use an extended set of Haar-like features (Figure 18), specially the rotated ones, to better detect vehicles positioned diagonally in the images.

Implementing the cascaded detector form may help in reducing false positives.

And of course, we could always try out pre-processing methods such as flattening the fish-eye image, to make training easy for the classifiers.

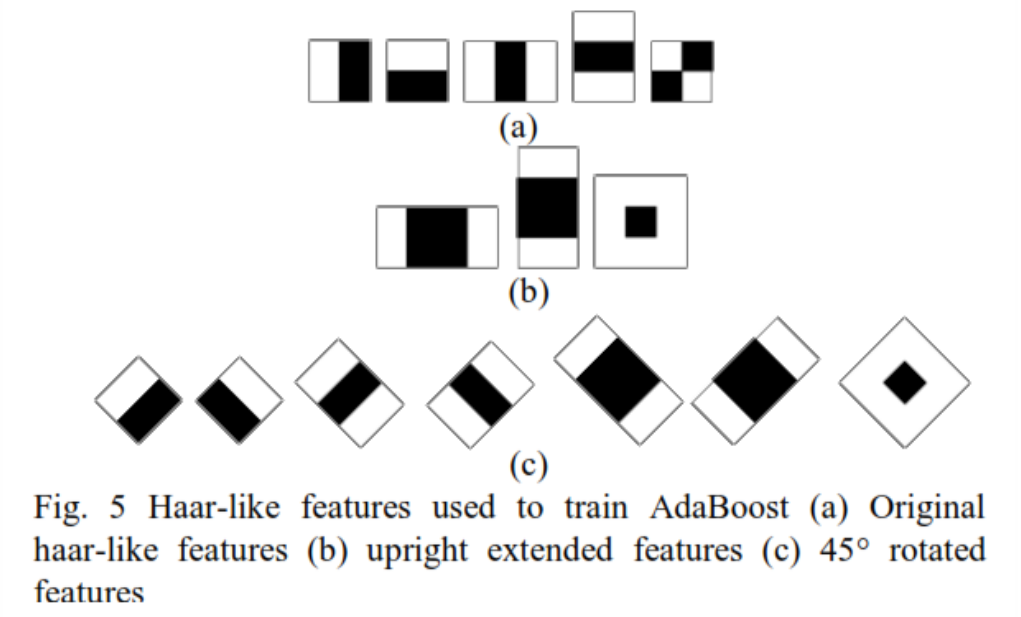


Figure 18: Examples of extended Haar-features (figure source: [3])

References

[1] H.-C. J. J.-H. Kim and J.-H. Lee. "improved adaboost learning for vehicle detection. *Int. Conf. on Machine Vision*, pages 347–351, Dec. 2010.

[2] D. C. Lee and T. Kanade. Boosted classifier for car detection. 2007.

[3] J.-M. Park, H.-C. Choi, and S.-Y. Oh. Real-time vehicle detection in urban traffic using adaboost. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3598 –3603, oct. 2010.

[4] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision (IJCV*, pages 137–154, 2004.

Acknowledgments

a. Thanks to Lt. Michael Evans of Hanover Police Department for providing the raw image database.
b. Thanks to Rushni Shaikh, Binghamton University and Valentin Siderskiy, Polytechnic Institute of NYU for helping in cropping and building the training dataset.
c. Thanks to Louis Buck for discussing Viola-Jones feature selection process with me.
d. Thanks to Prof. Torresani for helping me set up the project, suggesting strategies and offering helpful insights throughout the process.