Large-Scale Vehicle Detection in Challenging Urban Surveillance Environments

Rogerio Feris
IBM Research
rsferis@us.ibm.com

James Petterson
NICTA / ANU
jpetterson@gmail.com

Behjat Siddiquie
U. Maryland
behjat@cs.umd.edu

Lisa Brown
IBM Research
lisabr@us.ibm.com

Sharath Pankanti
IBM Research
sharat@us.ibm.com

Abstract

We present a novel approach for vehicle detection in urban surveillance videos, capable of handling unstructured and crowded environments with large occlusions, different vehicle shapes, and environmental conditions such as lighting changes, rain, shadows, and reflections. This is achieved with virtually no manual labeling efforts. The system runs quite efficiently at an average of 66Hz on a conventional laptop computer. Our proposed approach relies on three key contributions: 1) a co-training scheme where data is automatically captured based on motion and shape cues and used to train a detector based on appearance information; 2) an occlusion handling technique based on synthetically generated training samples obtained through Poisson image reconstruction from image gradients; 3) massively parallel feature selection over multiple feature planes which allows the final detector to be more accurate and more efficient. We perform a comprehensive quantitative analysis to validate our approach, showing its usefulness in realistic urban surveillance settings.

1. Introduction

Security incidents in urban environments span a wide range, starting from property crimes, to violent crimes and terrorist events. Many large urban centers are currently in the process of developing security infrastructures geared mainly to counter terrorism with secondary applications for police and emergency management purposes. In this context, the ability to automatically search for objects of interest, in particular vehicles, is extremely important. The recently introduced concept of "Searchable Video Analytics" [7] allows a system to answer queries such as "Show me all the two-door red vehicles in camera X from time Y to Z". A pre-requisite to enable this capability is to accurately locate vehicles in the video images, so that attribute extraction and indexing can be performed. In this paper, we address the problem of vehicle detection in real-world surveillance videos, although we believe our techniques are general and could be applied to other objects as well.

(a) (b)

Figure 1. Traditional methods based on background modeling fail to segment vehicles in common urban surveillance conditions. (a) A typical crowded scene. (b) Corresponding foreground blobs obtained through background subtraction. Note that groups of vehicles are clustered into the same blob.

Assuming a static surveillance camera monitoring an urban environment, our goal is to detect vehicles in each video frame captured by the camera. Urban scenarios pose unique challenges for vehicle detection. High volumes of activity data, different weather conditions, crowded scenes, partial occlusions, lighting effects such as shadows and reflections, and many other factors cause serious issues in real system deployments, making the problem very challenging. Traditional methods based on background modeling [16, 17] generally fail under these difficult conditions, as illustrated in Figure 1.

Overview of our Approach. We start by describing a method to automatically collect training samples for a particular camera view with almost no user interaction (Section 3.1). Based on very few manual exemplars (usually 1-4) provided by the user, a rule-based classifier using *motion and shape cues* is created to collect a large set of training samples in *low-activity* conditions with almost no false alarms. In a co-training scheme, we use this collected data to train a camera-specific *appearance-based detector*, which works in *high-activity* scenes and is robust to environmental factors such as different weather conditions, shadows, rain, etc.

This data collection technique was applied to capture training samples from many city surveillance cameras, leading to a gigantic dataset of real-world vehicle images. Our dataset contains nearly one million images, about 1000x the

size of existing publicly available vehicle datasets [1, 8]. In our work, we used this data to train per-camera detectors, but it could be used to train generic detectors as well.

In order to deal with partial occlusions, we propose a method for generating realistic occluded vehicles through Poisson image reconstruction from image gradients (Section 3.2). The key idea is to take advantage of the appearance and structure of vehicles occluding other vehicles. We quantitatively compare our method with other techniques, showing very good results in challenging crowded scenes.

Finally, we show that large-scale feature selection over multiple feature planes improves the accuracy as well the efficiency of the final detector (Section 3.3). We consider a huge set of potential feature configurations and develop a parallel version of Adaboost learning for making the feature selection problem tractable.

2. Related Work

Various models and methods have been proposed for appearance-based object detection, in particular vehicle detection. Examples include the seminal work of Viola and Jones [18] and many extensions using different features, such as edgelets [19] and strip features [20], as well as different boosting algorithms like Real Adaboost and GentleBoost. Support vector machines with histograms of oriented gradients have also been a popular choice for object detection [2, 5]. In earlier work, Schneiderman and Kanade [14] showed good vehicle detection results using statistical learning of object parts. Although appearance-based detectors have achieved very good performance in challenging scenarios, they usually require tedious labeling of thousands of training samples to work well. In addition, most methods run below 15 frames per second in conventional machines, which is not desirable for large-scale surveillance systems requiring many video channels to be processed by a single machine.

Co-training and online learning methods [11, 12] alleviate the manual labeling issue, while constantly adapting the detector as new data comes in. A common limitation of these techniques is the inaccuracy in capturing online data to correctly update the classifier. Differently, our method uses a simple combination of motion and shape cues to capture a diversified set of vehicles during extended periods of time without any false alarms.

Several datasets have been proposed for learning and evaluation of vehicle detection algorithms. Examples include the UIUC [1] and USC [8] datasets as well as generic object recognition datasets which include cars as an object category, e.g., Caltech [6], MSRC and the yearly Pascal VOC challenges [3]. However, these datasets mostly consist of images of vehicles restricted to frontal/rear and side poses and the number of images of cars is of the order of 1000, which in our opinion, is insufficient for capturing the entire degree of variation in the appearance of cars due to changes in pose, viewpoint, illumination and scale. Our new dataset consists of around 1 million images of vehicles in real-world urban surveillance settings, which should serve as a useful learning and evaluation resource for the computer vision community.

Methods for occlusion handling in object detection [9] generally rely on object part decomposition and modeling [5]. In our application, however, these methods are not well suited due to the low-resolution vehicle images. Video-based occlusion handling from the tracking perspective has been addressed by Senior et al [15], but it assumes objects are initially far apart before the occlusion occurs.

Large-scale learning is an emerging research topic in computer vision. Recent methods have been proposed to deal with a large number of object classes [13] and large amounts of data. In contrast, our approach deals with large-scale feature selection, showing that a huge amount of local descriptors over multiple feature planes coupled with parallel machine learning algorithms can improve not only the detector accuracy, but also its efficiency at test time.

3. Proposed Approach

In this section we describe our approach for automatic vehicle detection in urban scenes, including the training dataset formation, synthetic generation of occlusions, and large-scale detector learning.

3.1. Automatic Data Collection

Since it is infeasible to manually collect a large and diverse dataset consisting of cropped images of vehicles, we devised a simple procedure to automatically collect images of vehicles in traffic videos, which we now describe.

We collected videos from about 30 traffic surveillance cameras (Fig. 2a), which constantly captured data over a period of several months, providing us with data in a variety of illumination and weather conditions. Furthermore, due to variations in the positions of these cameras with respect to the direction of traffic, there were large variations in poses of vehicles captured from different cameras.

In each camera-view, vehicles usually appear in 1-4 poses, as a traffic camera typically overlooks either a single road or an intersection. We collect data per camera for each vehicle pose independently by following the steps below:

- We manually define one or more regions-of-interest (ROI), which specify the regions of the image from where we want to capture vehicles.

- In each ROI, we perform background subtraction [17] to obtain the bounding boxes of foreground blobs at each video frame. We also obtain the associated mo-

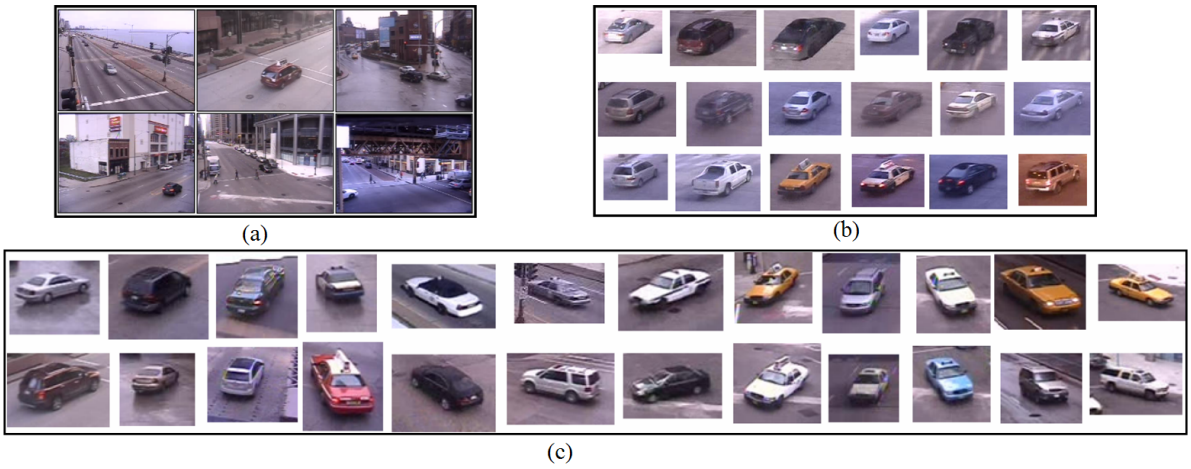

Figure 2. (a) Sample shots from some of the traffic surveillance cameras. (b) **Illumination Variation in the Dataset:** Images of cars collected from a single camera view showing the illumination variation present in the dataset. (c) Vehicles categorized into 12 categories, according to their direction of motion $\{0^\circ - 30^\circ, 30^\circ - 60^\circ, \ldots, 330^\circ - 360^\circ\}$, showing the diversity in pose of the cars in the dataset.

tion direction of foreground blobs through standard optical flow computation.

- We collect vehicles by using a simple rule-based classifier which analyses the shape and motion of foreground blobs at fixed time intervals. More specifically, in our implementation, we just check whether the *aspect ratio*, *size*, and *motion direction* of a particular foreground blob are within a pre-defined range of values that characterizes a vehicle. The range of values is obtained heuristically from very few manually labeled exemplars (usually 1-4, depending on the number of vehicle poses in the scene).

This simple procedure enables us to collect a large number of images of vehicles, while requiring minimal supervision. Figure 2b shows examples of training samples captured using a 10 hours video (from 8am to 6pm) for a specific camera and specific vehicle pose. In this experiment, we were able to capture few thousands of samples without false alarms. Note that we have *many false negatives as we are conservative*, e.g., in this video the classifier rejected many vehicle samples in crowded periods, or vehicles with long attached shadows, etc. However we can see in Figure 2b that we capture samples containing a huge amount of appearance variation - different lighting, weather conditions, vehicle models, etc. This is extremely important for training the appearance-based detector described in Section 3.3.

Using this data collection method, we were able to collect a dataset consisting of about 1 million images of vehicles. We categorized each vehicle into one of 12 different categories depending on its motion direction, to give a coarse estimate of its pose (Fig. 2c). There is a wide variation in the scale and pose of vehicles collected from different cameras, even when they have the same motion direction. We notice very few false alarms in the data collection process - in rare cases e.g., when a group of small objects have the same aspect ratio, size, and motion direction of a vehicle. These samples were manually pruned from our dataset.

3.2. Poisson Occlusion Handling

Although the algorithm described in the previous section can collect vehicle samples under significant variation of appearance, it fails to capture samples with partial occlusion. In this section we show a fully automatic process to generate realistic partially occluded vehicle images. Adding images with occlusions to the training set makes the detector much more robust to crowded scenarios, as we will show later in our experiments.

Figure 3 illustrates our algorithm. For a given vehicle image I_A, we randomly select another vehicle image I_B with its associated foreground mask M_B. We first dilate M_B to make sure the mask entirely contains the vehicle. Then we follow the steps below:

- Let I_{AB} be the image formed by pasting the vehicle of image I_B (i.e., the region defined by M_B) at a random location of image I_A.
- Compute the intensity gradient: $G(x, y) = \bigtriangledown I_{AB}(x, y)$
- Let $G'(x, y)$ be the modified gradient field obtained by adding zeros to $G(x, y)$ along the border pixels of the foreground mask M_B.
- Reconstruct image I'_{AB} which minimizes $|\bigtriangledown I'_{AB} - G'|$

Image reconstruction from gradients fields, an approximate invertibility problem, is still a very active research

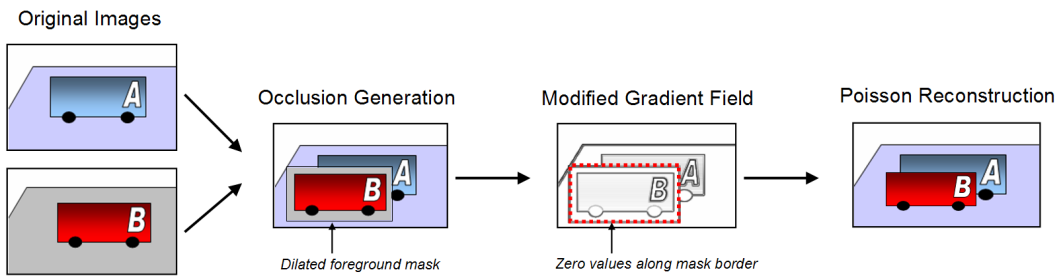

Figure 3. Synthetic generation of occluded vehicles based on Poisson image reconstruction from gradient fields.

Figure 4. Examples of realistic occluded vehicles generated by our algorithm. No user intervention is required.

area. In R^2, a modified gradient vector field G' may not be integrable. We use one of the direct methods proposed by Fattal et al. [4]. The least square estimate of the original intensity function, I'_{AB}, so that $G' \approx I_{AB}$, can be obtained by solving the Poisson differential equation $\bigtriangledown I'_{AB} = divG'$, involving a Laplace and a divergence operator. We use the standard full multigrid method to solve the Laplace equation. We pad the images to square images of size the nearest power of two before applying the integration, and then crop the result image back to the original size.

The resultant image I'_{AB} is added to the collection of training samples and this process is repeated so that many occluded vehicle samples are generated. Note that I'_{AB} provides a seamless blending of two vehicle images, even when they are captured under different lighting conditions. The superimposed image I_{AB} may contain pieces of the road and other artifacts due to noise in the foreground mask M_B. Since we modify the gradient along the border pixels of M_B,the resultant image I'_{AB} is much cleaner and realistic.

A similar algorithm is applied to generate images where the vehicle of image I_A (which is always positioned in the center of the image) is the occluder. Figure 4 shows examples of realistic training samples generated by our method. The process is fully automatic.

It is important to note that we always have a vehicle in the center of the training image. The position of the other vehicle is constrained by just allowing it to be placed at random locations that will create partial occlusions, not full occlusions. The training images are further cropped to have a tighter bounding box around the vehicle in the center. Since at test time our detector is based on sliding windows, if we have one vehicle occluding another vehicle, then the detector will fire twice, i.e., we will have two bounding boxes (one for each vehicle).

3.3. Large-Scale Detector Learning

For each camera-view, we learn a specific vehicle detector for each vehicle pose using training samples collected automatically as described in the previous sections. Therefore, at test time, we have usually 1-4 detectors running per camera, which are interleaved across the video frames to improve frame rate.

The basis of our learning algorithm is the framework proposed by Viola and Jones [18]. It consists of a cascade of Adaboost classifiers, where the weak learners are simple thresholds over Haar-like features. Each stage of the cascade is tuned to minimize false negatives at the expense of a larger number of false positives – this allows fast inference by quickly discarding background images. Bootstrapping is also employed by selecting negatives examples where the previous stages have failed. For details, see [18].

The key novelty of our learning algorithm is the introduction of multiple feature planes in the feature selection process, as shown in Figure 5. By considering feature planes such as red, green, and blue channels, gradient magnitude, Local Binary Patterns, and many others, we allow the final detector to be much more powerful, combining Haar-like features of different modalities. In this framework, feature selection is performed over a pool containing a huge set (potentially millions) of feature configurations. This poses a serious problem in terms of training time, as even with a single feature plane and a few thousand images Adaboost training takes days on a standard desktop machine. Therefore to deal with a huge pool of local feature

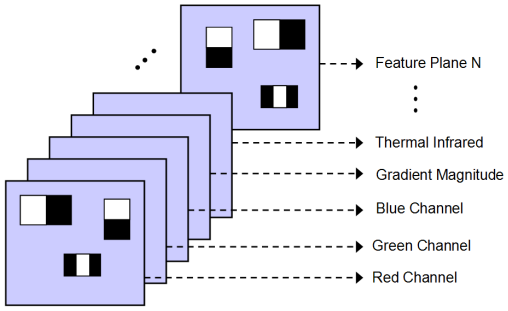

Figure 5. Feature selection over multiple feature planes. A feature pool containing a huge set (order of millions) of feature configurations is generated.

descriptors as we are proposing we need a way to parallelize training.

Adaboost is inherently sequential [10], making it difficult to scale in general, but in this particular setup there is a simple solution: parallelization at the level of features. At each step during training we have to compute a large number of features for all training images and select the one that better classifies the data; this can be done in parallel, with each CPU working on a subset of the features, and the amount of synchronization necessary is minimal: each CPU has to report only the best feature of its subset.

Additionally, at each stage a set of negative patches has to be selected from the set of available negative images. The selected patches are the ones for which the current classifier fails. This is the most time consuming activity in later stages of the cascade training, taking hours even for a small training set. Parallelization can also be implemented here, with each CPU searching for negative patches in a different subset of the negative images. Again, the amount of time spent on synchronization[1] here is comparatively very small, allowing for an almost linear speed-up with the number of CPUs employed.

So far in our implementation we have considered parallel feature selection over four color planes (gray-scale, red, green, and blue channels). As we will show in the experimental section, by adding color we not only improve the robustness of the classifier but also get a sparser solution, with a smaller number of selected features. That, in turn, reduces computation time during inference. We are currently adding more feature planes (gradients and texture descriptors, multispectral planes, etc.), which should improve results even more.

4. Experiments

Rather than training a general, single vehicle detector using our large dataset, we are currently training multiple specific detectors for each camera. Without loss of generality, we choose one camera for our quantitative analysis. We collected a challenging test set from this specific surveillance camera containing 229 images and 374 vehicles of a single pose (side-view). The images were captured in different months, covering different weather conditions including sunny and rainy days, different lighting effects, such as shadows and specularities, and different periods of time such as morning and evening. In addition, we split our test set into two groups: *high activity*, i.e., crowded scenes with many occlusions (104 images and 217 vehicles) and *low activity* (125 images and 157 vehicles).

We applied our data collection technique described in Section 3.1 to a 5 hours (from 2pm to 7pm) video sequence of the same camera but in a different day/month of the period used to capture the test images. This way we could collect 1800 training samples automatically without any false alarms, which were then re-sized to 26x10 resolution. A set of nearly 1000 negative images (non-vehicle data) were collected from the web and a cascaded Adaboost classifier based on Haar-like features was learned with a single gray-scale feature plane; later we will show the enhancement of learning with multiple planes. Figure 6a shows a couple of examples of training samples of this standard detector. At test time we apply the detector at different positions and scales, in the same sliding window scheme of [18]

Occlusion Analysis. In order to test our occlusion handling technique, we synthetically generated 1700 additional occluded vehicles using the technique described in Section 3.2 (Figure 6b) and added them to the training set to create a new detector. For comparison, we also trained a detector using images with occluders consisting of random noise (Figure 6c) and a part-based detector using images of the top part of the vehicles which are generally not occluded (Figure 6d). The ROC curves are shown in Figure 7. Note that our proposed approach significantly outperform all other techniques in crowded scenes, while having comparable performance to the standard detector in low activity scenes, which is reasonable. The occlusion noise detector does not take into account the appearance and structure of the occluder as in our approach. The part-based detector does not have good performance due to the low resolution of the vehicle images.

Large-Scale Feature Selection. We compared the standard Adaboost detector using one gray-scale feature plane with our approach using massively feature selection over multiple feature planes. As of now, we used four planes (gray-scale, red, green, blue channels). Figure 8 shows the ROC curves. In addition to achieving improved accuracy, our large-scale feature selection scheme allows the final detector to be more sparse and therefore more efficient (see Figure 10). Sample detection results of our quantitative analysis can be seen in Figures 9a and 9b. We have also applied the same process (automatic data collec-

[1] We used Message Passing Interface (MPI) in our implementation.

Figure 6. Examples of training images used in different experiments.

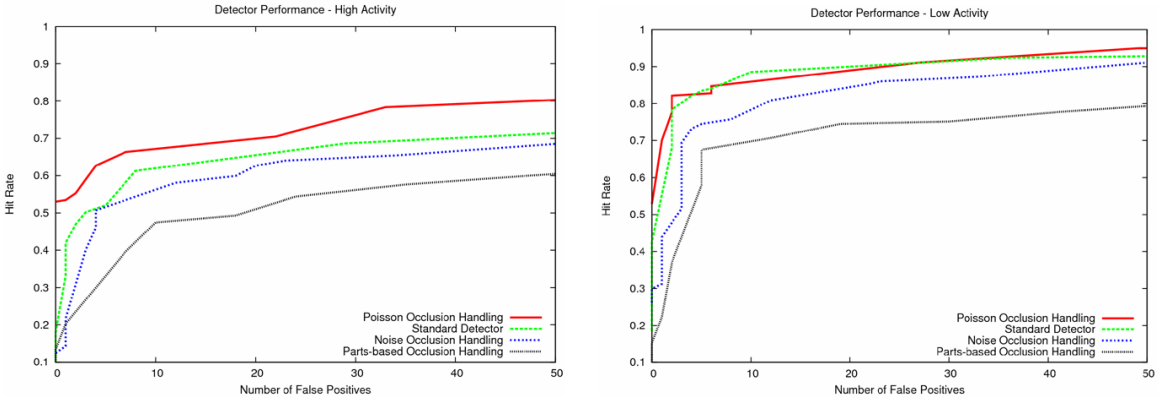

Figure 7. Our occlusion handling approach significantly outperform other methods in high activity scenes (left), while having comparable performance to the standard detector in low activity periods (right).

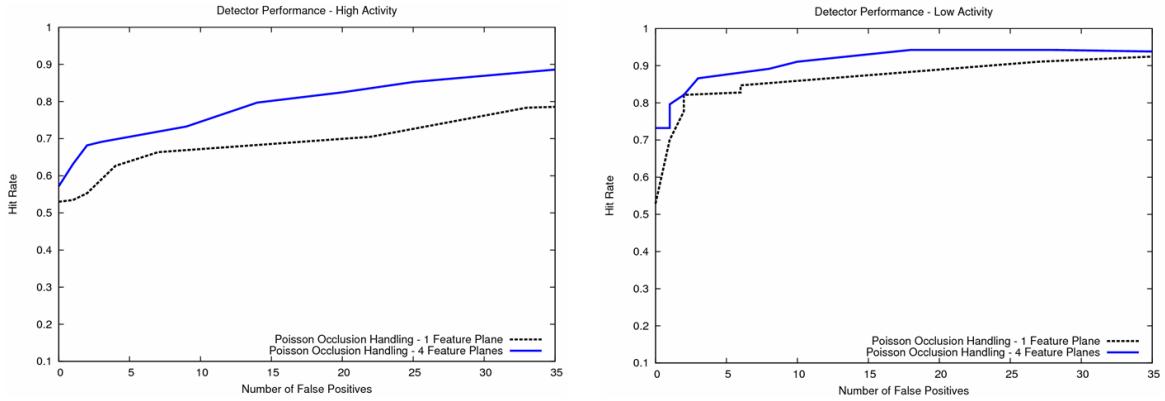

Figure 8. Comparison of our approach using massively parallel feature selection over multiple planes with standard Adaboost detection using a single gray-scale feature plane, in high activity (left) and low activity periods (right).

tion, occlusion generation, and detector learning) to other camera views, obtaining similar results. Qualitative results are shown in Figure 9c. Superior results can be obtained by applying our vehicle detector to video, by constraining the search process to foreground regions obtained through background modeling. We refer the reader to video demos at http://rogerioferis.com/demos.html. Our system runs quite efficiently, at an average of 66Hz on a conventional laptop computer (2.3GHz, 3GB of RAM).

Discussion. The reason why our method is robust to *environmental changes (shadows, rain, etc.)* is due to the fact that our data collection method captures many images under quite different lighting conditions, and also due to our rich feature pool. The robustness to *crowds and partial occlusions* come from our occlusion handling method. The *efficiency* of our approach is due to the fact that we are learning detectors using samples of the same camera, which leads to a classifier with much less features than a generic vehicle detector. In addition, as we showed above, we obtain even more sparsity with large-scale feature selection. Failure cases include vehicle images occluded by more than 40% and other examples for which the detector is not able to generalize. The performance could be improved even more by collecting more training data over different days.

Although our data collection technique can capture different vehicle classes, we can not detect large buses and bikes, for instance. As we run few pose-specific detectors per camera, we may not detect vehicles when they undergo poses not covered by the set of detectors, for example, when a car is turning. In our application, however, the detection output is used for indexing and to improve tracking, meaning that we need high precision but some false negatives can

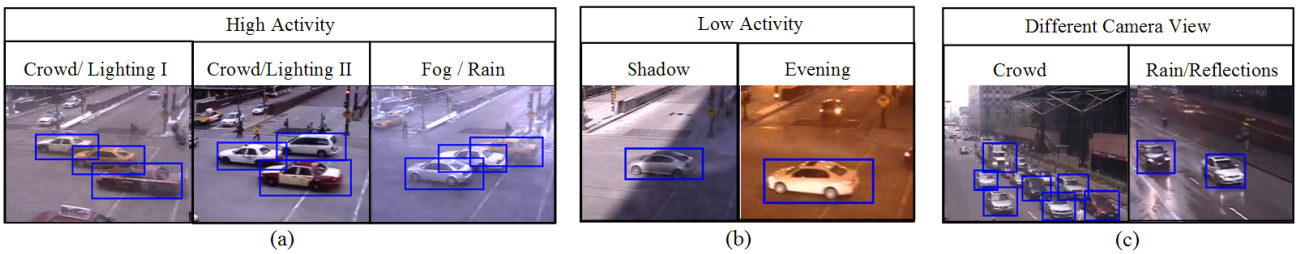

Figure 9. Detection results. (a) High activity and (b) Low activity samples used in our quantitative analysis. (c) Different camera view

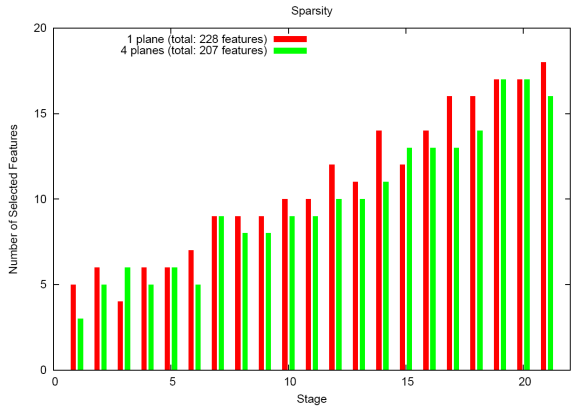

Figure 10. Number of selected features at each stage of the classifier. Training with more features yields a sparser solution.

be tolerated. Finally, we note that our occlusion handling technique takes into account the appearance of vehicles occluding other vehicles, but not other types of occlusions, such as a vehicle occluded by a lamp post. This could be done by automatically collecting data from object tracks in low-activity periods as vehicles may get occluded.

5. Conclusion

We have presented a new approach for vehicle detection in challenging urban scenarios, involving three main components: automatic training data collection, synthetic generation of occlusions, and large-scale feature selection. Future work include 1) exploiting our vehicle dataset to learn a generic vehicle detector using millions of images; 2) adding more feature planes to generate a feature pool containing hundreds of millions of local features; and 3) develop large-scale online adaptation methods.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on PAMI*, 26(11), 2004.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[4] R. Fattal, D. Lischinski, and M.Werman. Gradient domain high dynamic range. In *ACM SIGGRAPH*, 2002.

[5] P. Felzenszwalb and R. G. andD. McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010.

[6] G. Griffin, A. D. Holub, and P. Perona. The caltech-256. *Caltech Technical Report*.

[7] A. Hampapur, L. Brown, R. S. Feris, A. Senior, C. Shu, Y. Tian, Y. Zhai, and M. Lu. Searching surveillance video. In *AVSS*, 2007.

[8] C. Kuo and R. Nevatia. Robust multi-view car detection using unsupervised sub-categorization. *WACV*, 2009.

[9] Y. Lin and T. Liu. Fast object detection with occlusions. In *ECCV*, 2004.

[10] S. Merler, B. Caprile, and C. Furlanello. Parallelizing AdaBoost by weights dynamics. *Computational Statistics & Data Analysis*, 51(5):2487–2498, 2007.

[11] S. A. O. Javed and M. Shah. Online detection and classification of moving objects using progressively improving detectors. In *CVPR*, 2005.

[12] P. Roth, H. Grabner, D. Skocaj, H.Bischol, and Leonardis. On-line conservative learning for person detection. In *PETS Workshop*, 2005.

[13] O. Russakovsky and L. Fei-Fei. Attribute learning in large-scale datasets. In *ECCV 2010 Workshop on Parts and Attributes*, 2010.

[14] H. Schneiderman and T. Kanade. A statistical approach to 3D object detection applied to faces and cars. In *CVPR*, 2000.

[15] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. *Journal of Image and Vision Computing*, 24(11), 2006.

[16] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1998.

[17] Y. Tian, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. In *CVPR*, 2005.

[18] P. Viola and M. Jones. Robust Real-time Object Detection. In *International Journal of Computer Vision*, 2001.

[19] B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *ICCV*, 2007.

[20] W. Zheng and L.Liang. Fast car detection using image strip features. In *CVPR*, 2009.