Efficient 24/7 Object Detection in Surveillance Videos

Rogerio Feris  Russell Bobbitt  Sharath Pankanti  Ming-Ting Sun
IBM T. J. Watson Research Center, New York  University of Washington, Seattle
rsferis@us.ibm.com

Abstract

We address the problem of 24/7 object detection in urban surveillance videos, which presents unique challenges due to significant object appearance variations caused by lighting effects such as shadows and specular reflections, object pose variation, multiple weather conditions, and different times of the day. Rather than training a generic detector and adapting its parameters over time to handle all these variations, we rely on a large set of complementary and extremely efficient detector models, covering multiple overlapping appearance subspaces. At run time, our method continuously selects the most suitable detectors for a given scene and condition, using a novel approach inspired by parametric background modeling algorithms. We provide a comprehensive experimental analysis to show the effectiveness of our approach, considering traffic monitoring as our application domain. Our system runs at 100 frames per second on a standard laptop computer.

1. Introduction

In the past decade, significant progress has been made on visual object detection [16, 1, 6, 12], which is a key component of smart video surveillance systems. However, there are limited studies on analyzing the performance of object detectors in practical 24/7 surveillance scenarios. In fact, many challenges remain to be addressed in order to develop reliable detectors that run continuously over extended periods of time.

Most commercial systems rely on background modeling techniques [13, 15] for detecting moving blobs as a proxy for objects in the scene. These blob-based approaches are efficient and work reasonably well in low-activity scenarios. However, they are limited in their ability to handle typical urban conditions such as crowded scenes, where multiple objects are frequently merged into a single motion blob, compromising higher-level tasks such as object classification and extraction of attributes.

More recently, object-centered architectures which rely on appearance-based object detectors (e.g., pedestrian and vehicle detectors) have proven quite effective to replace or complement blob-based approaches [5]. Despite the extensive evaluation benchmarks and significant progress made in the field, existing off-the-shelf detectors still face significant challenges when deployed in 24/7 surveillance systems due to the wide range of appearance variations. As an example, the appearance of objects such as vehicles dramatically changes from daylight to night-time. Multiple weather conditions (rain, snow, ...), and lighting effects such as shadows and specular reflections also cause serious issues. Different camera placements and object pose variations are yet another source of dramatic changes in object appearance.

Online adaptation methods [7, 9, 10] have been proposed to automatically adapt a generic detector to different scenarios, but either they require a few labeled samples from the target domain or, in the case of unsupervised adaptation, they are sensitive to drifting. In addition, training a generic detector to handle a wide range of appearance variations usually requires more features, incurring computational costs that may not be tolerable for many applications.

In this paper, we address the problem of 24/7 object detection in surveillance videos by formulating adaptation as an online model selection mechanism. More specifically, rather than training a generic classifier and attempting to adapt its parameters to a wide range of conditions, we instead create a large set of extremely efficient detector models, covering multiple overlapping appearance subspaces, and at run time our method continuously selects the most suitable detectors for a given scene and condition. The suitability of detectors is measured based on a distribution of their calibrated confidence scores, which is updated over time. By thresholding this distribution, we classify each detector as foreground or background, where foreground detectors (suitable to the current condition) run much more frequently across the video sequence, whereas background detectors (not suitable to the current condition) run sporadically but are allowed to re-emerge as foreground detectors at a later time. Throughout this paper, we use the term background detector to denote an object detector that is not ap-
licable to the current scenario. This term should should not be confused with the scene background.

To the best of our knowledge, this framework is novel and has proven successful in several real-world deployments of our system worldwide. In addition to handling a wide range of appearance variations due to continuous 24/7 processing, our system runs at around 100 frames per second on a standard laptop computer (2.20 GHz, 8GB RAM). We consider vehicle detection for traffic monitoring as our application domain, although we believe our approach can be generalized to other objects as well.

2. Related Work

Appearance-based object detectors [16, 1, 6, 12] have shown to be very effective to segment objects in crowded scenes [5], but are not robust to the wide range of appearance variations inherent in systems that need to run over long periods of time. Recently, deep convolutional neural networks have achieved breakthrough results in image classification [11] and object detection [6, 12]. A major advantage of these neural network models is that features are shared across categories and learned to form a powerful hierarchical representation suited to the task at hand. However, these models require significant more computational processing than our approach and may not be applicable to deployments that require processing many channels per server. We believe our approach is complementary and could be used in a sensitive mode to generate region proposals for deep neural networks.

We classify detectors as “foreground” or “background” using a method inspired by the work of Stauffer and Grimson [13]. In their work, a set of $K$ Gaussian models is maintained for each pixel, and an online update rule is used to determine the weights and parameters of the Gaussians at any time period. The Gaussians are then sorted according to the weights and a threshold is applied to classify each Gaussian as foreground or background. In our approach, we maintain a set of $K$ detectors and use a similar online update rule to determine their weights and a similar thresholding scheme to classify them as foreground or background. We note, however, that our work operates at the detector level instead of the pixel level, and has a very different goal compared to [13].

Our selection mechanism is also related to approaches based on mixture of experts [8], where each expert is focused on a particular portion of the object appearance manifold and a classifier is used to predict which expert should be used for any given input. Different from our approach, a model needs to be trained beforehand for expert selection, hindering scalability when a large number of experts is considered, and also adding another reasoning module per frame, incurring more computational cost. Our framework is scalable, and any off-the-shelf detector can be added to the pool of detectors without requiring an extra learning step. In our method, the computational cost is independent of the number of detectors.

3. Technical Approach

Our approach consists of two stages: (i) a training stage where a large set of complementary and extremely efficient detectors is generated (offline), and (ii) an adaptation mechanism based on online model selection (run time). Next we describe these two stages.

3.1. Learning Efficient Detectors

We follow the work of Feris et al [3] to capture a large set of vehicle images from around 30 surveillance cameras without significant annotation cost. Our training dataset contains around one million images, covering many different weather conditions and times of the day, and a wide variety of vehicle poses. As in [3], this data is partitioned into motionlet clusters, i.e., clusters containing vehicles with similar motion direction, which is automatically computed using optical flow. Naturally, there is a strong correlation between these clusters and the pose of vehicles. In order to obtain better performance in crowded scenes, which often depict vehicles with partial occlusions, we augment this data with synthetic occlusions, using the approach described in [4], where Poisson image editing is used to synthesize vehicles occluding other vehicles.

For each motionlet cluster, we further subdivided the data based on attributes such as daylight, night-time, and weather condition, and trained complementary detectors [2] covering multiple appearance subspaces. The detectors are based on cascades of GentleBoost classifiers using Haar-
We denote foreground detectors as those that are suitable for a given scene and condition, and background detectors as those that are not suitable. We note again that the term background used here should not be confused with the scene background. We continuously classify all detectors in the pool as either foreground or background by thresholding a time-varying weight distribution based on the detector calibrated scores as described next.

Initially, during the system start-up, all models in the pool are initialized as background detectors. We interleave the detectors across the video sequence by running detector \( D_k \) at frame \( t \), where \( k = t \mod N \), and \( N \) is the total number of detectors in the pool. In other words, we run only one detector per frame in a round-robin fashion.

Each detector \( D_k \) has an associated weight \( w_{k,t} \) which indicates its suitability to the input video at time \( t \). Initially the weights of all detectors are equal and set to \( 1/N \). When a detector fires, the weight distribution is updated as follows:

\[
\begin{align*}
    w_{k,t} & = w_{k,t-1} + \alpha(M_{k,t} - w_{k,t-1}) \\
\end{align*}
\]

where \( M_{k,t} \equiv 1 \) for the detector that fired and 0 for all others, and \( \alpha = C_k \theta \) is the product of the detector confidence score \( C_k \) and the learning rate \( \theta \), which determines the speed at which the distributions parameters change. In our implementation, we set \( \theta = 0.001 \). After this approximation, the weights are renormalized. Similar to [13], \( w_{k,t} \) is effectively a causal low-pass filtered average of the thresholded firing probability of detector \( k \) from time \( 1 \) through \( t \). This is equivalent to the expectation of this value with an exponential window on the past values.

At any given time \( t \), all detectors \( D_k \) with associated weight \( w_{k,t} > T \), where \( T \) is a threshold, are classified as foreground detectors and the remaining ones are classified as background detectors. Generally only a few detectors are selected as foreground at a given time period. We process the video frames globally, but this selection process could also be applied to spatial regions.

3.2.2 Adaptive Interleaving

We have shown how to determine suitable detectors for a given camera and time period. We now show how to use
4. Experimental Analysis

We provide a comprehensive experimental analysis to show the effectiveness of our framework. We start by analyzing 24/7 adaptation from a single camera, in different time periods and weather conditions. Then we show that our adaptation approach offers superior performance over traditional approaches based on background modeling for continuous 24/7 processing, considering a wide variety of surveillance cameras.

4.1. Adaptation Analysis: Environmental Conditions and Times of the Day

We recorded video from a single surveillance camera during several days and identified subclips with different weather conditions such as cloudy, sunny, heavy rain, lighting effects such as specular reflections and shadows, and different times of the day such as evening and night. Figure 4a shows example frames depicting these conditions. For each subclip, we manually annotated 500 frames after a period of 5 minutes reserved for adaptation. The annotation consists of bounding boxes for all vehicles in the scene, around 5000 boxes in total.

The plots in Figure 3 show the adaptation weights for each detector in the pool, in conjunction with the corresponding hit rate and number of false alarms per image (FPPI). The indexes of the selected foreground detectors for the corresponding time period are also shown in the figure. Notably, high adaptation weights are clearly correlated with high hit rate and small FPPI, meaning that we are selecting the “right” models as foreground detectors. There is one exception which happens in the “heavy rain” plot in which detector 24 (night-time detector) is selected as foreground and does not offer the best hit rate/FPPI trade-off.

We inspected this video and identified an interesting explanation: this video is a transition from night to day, meaning that the night-time detector was correctly selected as foreground prior to the annotation. We noticed that the annotated frames correspond to the period in which detector 24 is decreasing its weight, and the other “correct” daylight detectors are increasing their weights, eventually becoming foreground detectors. As in [13], the time for adapting to these transitions is directly related to the learning rate parameter.

Another interesting observation is that some detectors may work well for different conditions but not for others, e.g., detector 7 works well for “cloudy” and “evening”, but does not work well for the other conditions. When we deploy our system to different cameras, there is a significant variation on the selected foreground detectors, due to object pose and lighting. This is an interesting characteristic of our approach which continuously select the suitable detectors for any given camera or environmental condition.

4.1.1 Counting at Night-Time

The appearance of vehicles changes completely from day to night. We designed a generic vehicle detector that attempts to handle these dramatic changes in appearance and compared to our approach. The generic detector model is similar to the specialized models used in our approach, except for the training data which contains much more variations. As an example, we tested both approaches at a night-time video clip for a vehicle counting application. At a similar operating point, with less than 10 false alarms for the entire video, our method detects 390 vehicles compared to 241 vehicles detected by the baseline. When combined with tracking for a counting application, the counting ratio with respect to the ground-truth is 1.043 for our method and 0.618 for the baseline. We note that generic detectors are also generally more computationally expensive, as more features are required to deal with the wide range of appearance variations.

Because we rely on online model selection, our approach is not sensitive to drifting, which is a common issue in self-adaptation methods. Compared to background modeling techniques, our approach detects objects instead of blobs. Figure 5 shows an example where blob-based approaches fail to segment objects due to headlights and reflections.
Figure 3. Analysis of our proposed adaptation framework considering multiple environmental conditions and times of the day. The plots show the adaptation weights for each detector, in conjunction with the corresponding hit rate and number of false alarms per image (FPPI). The circles and the numbers indicate the foreground detectors and their indexes, respectively.

4.2. Multiple Cameras: 24/7 blob-based versus object-centered analytics

Our complete system combines detection with tracking to generate a single keyframe per moving object in videos captured by surveillance cameras. We tested our complete system on 24 different traffic cameras, covering a wide range of object poses, lighting, and different resolutions (Figure 4b). For each camera, we randomly sampled a 20min video clip and annotated vehicle bounding boxes after the first 5min reserved for adaptation. During the benchmark process, we compared the output keyframe bounding boxes with the ground-truth bounding boxes to measure precision and recall.

We report a recall/precision of 0.71/0.92 for our system compared to 0.66/0.59 for the same system but with the object detector component replaced by a method based on background modeling [14]. This shows that object-centered analytics can be superior to blob-based approaches for 24/7 video processing, which is not an obvious conclusion due to the wide range of appearance variations that are present when a system needs to continuously run during extended periods of time.

4.3. Efficiency Analysis

Our approach for 24/7 object detection runs at about 100 frames per second on a standard laptop machine (2.20 GHz, 8GB RAM). This is critical for surveillance systems that need to process many channels per server.

5. Conclusion

We have presented a novel approach for 24/7 object detection in surveillance videos. Our method relies on a large set of extremely efficient detectors covering multiple overlapping appearance subspaces, which are continuously selected to fit the given scene and environmental condition, adapting to a wide range of appearance variations. Our system has been incorporated into an actual product and used with success in several deployments worldwide. Most previous works in appearance-based object detection have focused on specific benchmarks without covering the setting where an object detector needs to run continuously over extended periods of time. We hope our work will inspire others to pursue this direction, which is critical for real-world smart surveillance systems.

As future work, we plan to expand the number of detectors in our framework, focusing on the mis-detections of the current set, in order to fill up the holes of the object appearance manifold.
Figure 4. Our evaluation data: (a) Example video frames of a single camera, covering multiple environmental conditions and times of the day. (b) Video frames from multiple cameras, randomly sampled from different times of the day. Note the variation in lighting, object pose, and resolution. The images were intentionally pixelated to hide details about the location where the videos were captured.

References


