Training a Scene-Specific Pedestrian Detector Using Tracklets

Yunxiang Mao  
Zhaozheng Yin  
Department of Computer Science  
Missouri University of Science and Technology, USA  
ym8r8@mst.edu, yinz@mst.edu

Abstract

A generic pedestrian detector trained from generic datasets cannot solve all the varieties in different scenarios, thus its performance may not be as good as a scene-specific detector. In this paper, we propose a new approach to automatically train scene-specific pedestrian detectors based on tracklets (chains of tracked samples). First, a generic pedestrian detector is applied on the specific scene, which also generates many false positives and miss detections; second, we consider multi-pedestrian tracking as a data association problem and link detected samples into tracklets; third, tracklet features are extracted to label tracklets into positive, negative and uncertain ones, and uncertain tracklets are further labeled by comparing them with the positive and negative pools. By using tracklets, we extract more reliable features than individual samples, and those informative uncertain samples around the classification boundaries are well labeled by label propagation within individual tracklets and among different tracklets. The labeled samples in the specific scene are combined with generic datasets to train scene-specific detectors. We test the proposed approach on three datasets. Our approach outperforms the state-of-the-art scene-specific detector and shows the effectiveness to adapt to specific scenes without human annotations.

1. Introduction

In the past decade, a lot of appearance-based pedestrian detection methods were proposed to improve the performance of pedestrian detection [4, 5, 7]. Various features were explored to better describe positive and negative samples. Thousands of samples were acquired to build large training datasets in order to cover a large variety of viewpoints, lighting conditions, pedestrian poses, image resolutions, and backgrounds.

However, it is still very difficult to develop a single generic detector to handle all different scenarios. It has been shown in [4] that when applying a generic detector to a specific scene which is not the same as the dataset the generic detector is trained from, its performance drops significantly. For example, Fig.1(a) shows some samples from (1) the generic INRIA dataset [2] which mainly captures pedestrians in frontal view and (2) a specific scene [15] with a slight top-down view. After training a generic detector based on the INRIA dataset and applying it to the specific scene, false positives and miss detections happen frequently in the video frames (Fig.1(b)).

In fact, for videos taken by stationary surveillance cameras, the variations of viewpoints, lighting conditions, and backgrounds are limited within the context. Thus, the varieties of both positive and negative samples are much smaller than those in generic datasets. It is intuitive to add samples in the specific scene to train a scene-specific detector.

Moreover, a traditional way to train a detector is manually labeling positive and negative samples. It is acceptable to process a few scenes by manually annotating positive and negative samples from images. But when dealing with a large number of different scenes, it requires a lot of human efforts. A desirable way to solve this problem is to automatically acquire positive and negative samples from specific scenes for the scene-specific detector training.

1.1. Related Work

Recently, quite some research works focus on reducing the labor cost of manually labeling samples from specific scenes [12, 14, 1, 15]. How to correctly label samples
from the scene is critical for automatically training a scene-specific detector.

Ali et al. [1] train an appearance-based model built with the construction of object trajectories from tracking. They initialize the procedure of iterative training by manually cropping some samples, thus not completely automated. To automatically label samples, Rosenberg et al. [12] use self-training based on background subtraction to label scene samples. Only samples with high confidence scores are included in new training dataset. Co-training has been studied in [10, 13]. Multiple detectors based on different types of image features are used to label samples. The prediction result of one detector is used to train another detector. But co-training needs detectors to be independent, which is hard to achieve [3].

Scene contexts have been explored to improve the performance of pedestrian detector. Wang et al. [15] integrate multiple cues including motion, path models, locations, sizes and appearance to select positive and negative samples from specific scenes. Roth et al. [13, 14] utilize local context information to split the task of object detection into several sub-tasks. Samples appearing in the same location are classified into positives and negatives based on background subtraction, which may cause detector drift when stationary pedestrians are labeled as negative samples.

1.2. Motivations and Contributions

- **(1)** How to correctly classify unlabeled samples from a specific scene is a critical issue in training a scene-specific detector. Different from existing works which perform labeling on individual samples one by one, we propose to solve this problem by labeling tracklets which contain multiple samples from the same object over time. In our work, pedestrian tracking is considered as a data association problem on samples detected from pedestrian detection. We exploit multiple features for labeling tracklets. Thus, *labeling is propagated within individual tracklets.*

- **(2)** How to label uncertain samples in a scene is important to train a discriminative classifier (detector) since these uncertain samples around the margin are very informative. We build both positive and negative sample pools from samples in the confidently labeled tracklets. Uncertain samples are labeled only if they are close to one of the positive and negative sample pools but far away from the other sample pool. Hence, *labeling is propagated between labeled tracklets and uncertain tracklets in a scene.*

Fig. 2 shows the idea of label propagation within and between tracklets. Tracklet features are extracted on tracklet 1 and its samples are labeled (Fig.2(a)). While tracklet 2 is uncertain, it is compared with samples in other tracklets (Fig.2(b)). Once a sample in tracklet 2 is labeled, its label is propagated to other samples in tracklet 2 (Fig.2(c)).

2. Methodology

2.1. Overview

The diagram of proposed framework is shown in Fig.3. First, a generic detector is trained on a generic dataset. Given videos taken by stationary cameras in specific scenes, the generic detector is applied to detect pedestrian candidates in each individual image, which may include a lot of false positives and miss detections. Then, we track the detected candidates between consecutive frames to obtain reliable object trajectories (tracklets). Five features are explored to label tracklets: *travel distance* and *travel time* of each tracklet; the most possible *object size* for each location in the specific scene; *motion* information among samples in the same location across nearby temporal frames; *confidence scores* of samples in tracklets. Based on these features, some tracklets in the specific scene are labeled as positive or negative tracklets confidently while the rest are uncertain. We further label the uncertain tracklets by comparing them with samples from confidently labeled tracklets. The labeled tracklets are combined with the INRIA dataset to train a scene-specific detector. The procedure is repeated until there is no significant difference between two consecutively trained scene-specific detectors. Note that, the generic detector is used only once in the first iteration for initialization. The scene-specific detector is automatically and iteratively trained.

In this diagram, a tracklet-based framework is proposed to conquer the problem of labeling scene-specific samples detected by the current detector. We group detected samples which belong to the same object into tracklets based on their temporal and spatial information as well as their feature distance. The labeling process labels tracklets instead of individual samples, which is very effective to remove false positives and recover miss detections because tracklets have richer sets of features compared to individual samples and labeling is propagated in tracklets and between tracklets, i.e., informative samples around decision boundaries can be well-labeled.
2.2. Generic Detector Training

The proposed framework starts with a generic detector using the well-known Support Vector Machine (SVM) + Histogram of Oriented Gradient (HOG) [2] trained on the INRIA dataset. The generic detector is able to detect some pedestrian samples in the specific scene, but with many miss detections and false positives which are samples around decision boundaries and difficult to classify. To improve the detector performance, we need to correctly label these samples to adapt the generic detector to the specific scene.

2.3. Tracklet Generation

Based on the detection results we are able to track/link them into tracklets (a tracklet is a chain of samples belonging to the same object over time) by adopting the multi-object tracking method in [11]. Note that we do not need the performance of tracking to be perfect to conquer challenges like long-time occlusion. This association-based tracking can provide us satisfactory tracklets for labeling process later.

2.4. Tracklet Features

The key of online training system is that new unlabeled samples must be correctly labeled into the training dataset, otherwise the system tends to encounter the problem of drift, i.e., wrongly labeled samples in the training dataset will gradually decrease the detector performance by generating more and more mislabeled samples. During online training, the feedback from the current detector itself has been used to label new samples [10] to train the detector in the next iteration. But, it is hard to set a suitable threshold on the feedback score to determine which sample is confidently positive or not. If the threshold is too conservative, we lose some important informative samples which appear around the margin between positive and negative datasets. If the threshold is too aggressive, we take the risk of drift since more wrongly labeled samples may be included into the training dataset.

Since training a scene-specific detector only based on confidence score information from the current detector may result in decreasing performance of trained detectors, we explore more features in the following subsections. Furthermore, our features are extracted from tracklets (chains of tracked samples), which are more reliable than features of individual samples.

2.4.1 Travel Time

Travel time of a tracklet $T_i$ is defined as the number of samples in this tracklet, $N_i$. For example, a persistently-tracked pedestrian has a long tracklet (i.e., long travel time) while evanescent false positives have short tracklets (i.e., short travel time).

2.4.2 Pedestrian Size

When the viewpoint of a given scene is fixed, the size of a pedestrian in a specific location should be fixed, which is a strong cue to exclude false positives. The process of our size estimation is illustrated in Fig.4. First, the generic detector detects many samples (pedestrians or false positives) with true or wrong sizes in the scene over time, as shown in Fig.4(a). We use sizes of samples with high confidence scores to represent the right sizes for those locations (Fig.4(b)). For locations without size information, interpolation by Delaunay triangulation is used to estimate sizes of pedestrians within each triangle (Fig.4(c), where the vertex locations of the Delaunay triangles have their size information from Fig.4(b)). Thus, we get the most possible size for every location in the scene (Fig.4(d)). The size of an actually detected pedestrian sample at one location may not be exactly the same as our estimated size because of the variation of pedestrian heights in the image and the effect of rectangle clustering during sliding-window-based pedestrian detection. Thus, we allow a variance on the size estimation at each location. Define the
2.4.4 Motion

For a positive tracklet in which a pedestrian is moving, we can observe motion between consecutive samples due to body movements and for a negative tracklet which is from background, each sample of it contains much less observable motion. Therefore, motion in each sample of a tracklet can provide us a good feature to label tracklets. However, when a pedestrian stay in one location for a long period, there may be no motion in the samples of that tracklet, while samples from background can have temporary motion distraction, e.g., a pedestrian passes by that background (Fig.6). Thus, how to extract and represent motion information of a tracklet is important.

For any sample in a tracklet, we acquire more samples (e.g., over a few minutes) which appear in the same location before the start and after the end of that tracklet, taking these samples as references. Each sample of the tracklet subtracts itself by its references and compute the number of moving pixels. Median filtering is used to deal with noise. We denote the maximum number of moving pixels in the \( j_{th} \) sample of tracklet \( T_i \) as \( N_{i,j,m} \), and the number of pixels in one sample as \( N_{i,j,p} \). Then, we concatenate all the percentages \( (N_{i,j,m}/N_{i,j,p}) \) into one motion feature vector for tracklet \( T_i \).

2.4.5 Confidence Score

Confidence scores are not reliable to label individual samples because it is hard to set a suitable threshold to classify positive and negative samples, however we can adopt a “Winner-Take-All” strategy to represent a tracklet’s confidence score. We extract the highest confidence score of all...
samples in a tracklet as a feature for that tracklet. A tracklet is a chain of samples with Markov properties (neighboring samples in the chain are very similar), once there is a high confident sample in the chain, the confidence is propagated sequentially to other samples in the chain so the entire tracklet is considered as having a high confidence score.

For example, Fig. 7 shows two tracklet samples and their corresponding confidence scores. The pedestrian tracklet has little body motion over time and some individual samples of it have low confidence scores. But the samples are well-tracked hence the confidence score of the entire tracklet is still high due to the score propagation. As a comparison, a long tracklet continuously detected on a background has low confidence scores on samples and the confidence of this tracklet is low.

Figure 7. Confidence scores of samples in stationary tracklets. The confidence scores are changing with the little pedestrian movement or the lighting variations.

2.5. Tracklet Labeling

2.5.1 Positive Tracklet Labeling

The process to determine if a tracklet is positive or not is described in Fig. 8. First, to be labeled as positive, most samples of the tracklet should fit the correct sizes of their locations. Then we check whether the tracklet contains any sample with high confidence score larger than a score constraint $C_{\text{score}}$. Any sample with high confidence score is labeled as a positive sample and the same label is propagated to other samples in the tracklet. If there is no confident sample, we consider the travel time feature. To avoid labeling false positive tracklets from moving objects other than pedestrians, we require travel time to be larger than a time constraint $C_{\text{time}}$. If the tracklet satisfies the time constraint, we finally check its travel distance. Only tracklets with travel distance ($D_i$) larger than a travel distance constraint ($C_{\text{dis}}^i$) are labeled as positives.

2.5.2 Negative Tracklet Labeling

If a tracklet cannot be labeled as positive tracklet, then we check if it can be labeled as negative tracklet (Fig. 8). First, we check whether samples in one tracklet satisfy the size constraint or not. Since the size estimation provides us correct size for each location, samples with wrong sizes are highly probable to be negative. When all samples in a tracklet are with wrong sizes ($N_{i,s} = 0$), we label the tracklet as negative.

If we cannot label one tracklet by size, we use the travel distance and motion features of tracklet. Tracklets from background should stay in one location thus its travel distance $D_i$ should be smaller than $C_{\text{dis}}^i$, the constraint on the small travel distance. Furthermore, motion of samples in background tracklet should be small. To label a tracklet as negative we require that the tracklet does not contain any sample with $N_{i,j,m}/N_{i,j,p}$ larger than $C_{\text{motion}}$, the constraint on percentage of motion pixels.

2.5.3 Uncertain Tracklet Labeling

By the above rule-based classification, most of tracklets can be labeled as either positive or negative. But the remaining uncertain tracklets are also important to our scene-specific detector since most of them appear around the margin between positive and negative datasets. In order to label more uncertain tracklets, we compare samples in HOG space. We build positive and negative sample pools consisting of samples from both INRIA dataset and newly labeled tracklets. Since the number of new labeled samples in the current iteration is very large, we spatially uniformly sample labeled positive samples in the scene. For each location, only the sample with the highest confidence score is added to the positive sample pool. For stationary negative tracklet, only one sample is added to the negative sample pool while all moving negative samples are added. Given an uncertain tracklet, we calculate the Euclidean distance between each sample in the tracklet with all the labeled samples in the positive and negative sample pools. To label an uncertain sample, we require it to be far away from one of the positive and negative sample pools, but near the other one. Otherwise, it remains uncertain. Once a sample in an uncertain tracklet is labeled, its label is propagated to other samples in the same tracklet.
2.6. Scene-specific Detector Training

After tracklet labeling, we obtain a set of new labeled tracklets from the scene. For a positive tracklet, since the poses of a pedestrian in samples of adjacent frames vary slightly, we uniformly extract samples from positive tracklets into positive dataset. Only one sample in one negative tracklet is added to negative dataset. We combine the scene-specific positive and negative datasets with the INRIA dataset for scene-specific detector training to prevent the detector from drift. We iteratively train the scene-specific detector until there is no new samples available for training.

3. Experimental Results

3.1. Datasets

In the experiments, we evaluate our proposed approach on three datasets. The first one is the public CUHK Square data set [15]. The other two datasets are videos acquired by ourselves in front of a library in two different views.

The public CUHK Square dataset (Fig.9) is a 60-minute long video taken by a stationary camera from a bird-view, in which there are moving vehicles and many stationary pedestrians. One hundred frames are uniformly sampled from the last 30 minute videos with manual labeled ground truth. The first half of the video is used for training.

Each of the library datasets (Fig.11) contains a 60-minute long video. Occlusion happens in both datasets. The ground is not even, introducing difficulties for size estimation. In the library dataset 2, lighting condition changes a lot. We also use the first half of each video for training and the second half of videos for testing.

3.2. Evaluation Metric and System Parameters

As defined in PASCAL [6], a detected sample is correct if the area of the intersection between the detection window and the ground truth exceeds 50 percent of their union area. Recall rate versus False Positive per Image (FPPI) is employed as our evaluation metric. The detection rate is defined as the recall rate when FPPI is 1.

The parameter $C_{\text{size}}$, which requires sizes of most samples in one positive tracklets to be correct, is set to 0.9 to be conservative. $C_{\text{motion}}$ is set to be 0.1. Travel distance, travel time and confidence score of all tracklets are estimated as Gaussian distributions. $C_{\text{dis}}^2$ and $C_{\text{dis}}^5$ are set at where the cumulative distribution function of the travel distance Gaussian distribution equals to 10% and 50%. $C_{\text{time}}$ is estimated at where the cumulative distribution function of the travel time Gaussian distribution equals to 20%, and $C_{\text{score}}$ is determined at 80% of that of confidence scores.

3.3. Comparison on the CUHK Square Dataset

We compare our proposed framework with Wang et al. [15] on their CUHK square dataset. The evaluation results are shown in Fig.10. Our scene-specific detector significantly outperforms the generic detector and our detection rate outperforms that of [15] by 10 percentage points. In [15], when they train the specific detector, samples with high confidence scores are given large weights and samples with low confidence scores are given small weights. However, features of individual samples they use are not reliable which may result in wrongly labeled samples. They reduce the bad influence of wrongly labeled samples on training by giving them small weights close to zero based on visual comparison with other samples. But those wrongly labeled samples are very important indeed since they are near the margin between positive and negative datasets. We exploit different features in tracklets to robustly give samples correct labels, which largely avoids the problem of wrongly labeling and does not ignore important samples.

3.4. The Convergence of Iterative Training

We also evaluate the convergence of our framework on two datasets with different viewpoints and lighting conditions. As shown in Fig.12 and Fig.13, the scene-specific detector training converges after two iterations on both datasets. The detection rates of our scene-specific detector outperforms that of the generic detector by 32 percentage points and 28 percentage points, respectively. Fig.11 shows some sample detection results by scene-specific detector trained at different iterations. As the scene-specific detector is iteratively trained, there are less false positives and miss detections.
Figure 11. Sample detection results by our scene-specific detector trained at different iterations (from left to right). Top and bottom rows are library dataset 1 and 2, respectively.

Figure 12. The convergence of our scene-specific detector on Library Dataset 1.

Figure 13. The convergence of our scene-specific detector on Library Dataset 2.

4. Conclusion

In this paper, we proposed a new approach to automatically train scene-specific pedestrian detectors based on tracklets. The framework starts with a generic detector. By multi-pedestrian tracking, we link detection samples into tracklets. Reliable tracklet features are explored to label tracklets into positive, negative and uncertain ones. Labeling is propagated to uncertain tracklets based on comparison between uncertain tracklets and labeled tracklets. Experiments on three datasets show that our proposed approach is effective to train a scene-specific detector with better performance than a generic detector.

Acknowledgement

This work is supported by the National Science Foundation under Award Number CNS-1205695. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References