Object Detection in Surveillance Video from Dense Trajectories

by

Mengyao Zhai

B.Sc., Hebei University, 2013

Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

in the
Department of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Fall 2015

Approval

Name:	Mengyao Zhai
Degree:	Master of Science (Computer Vision)
Title:	*Object Detection in Surveillance Video from Dense Trajectories*
Examining Committee:	**Dr. Richard Vaughan** (chair) Associate Professor

Dr. Greg Mori
Senior Supervisor
Professor

Dr. Ze-Nian Li
Supervisor
Professor

Dr. Mark Drew
Examiner
Professor

Date Defended: 07 Dec 2015

Abstract

Detecting objects such as humans or vehicles is a central problem in surveillance video. Myriad standard approaches exist for this problem. At their core, approaches consider either the appearance of people, patterns of their motion, or differences from the background. In this paper we build on dense trajectories, a state-of-the-art approach for describing spatio-temporal patterns in video sequences. We demonstrate an application of dense trajectories to object detection in surveillance video, showing that they can be used to both regress estimates of object locations and accurately classify objects.

Keywords: object detection; dense trajectory; Gaussian process; regression

Acknowledgements

First and foremost, I want to thank my supervisor Dr. Greg Mori. He is the best supervisor I can ever imagine. I want to thank him for his patience, support and immerse knowledge. He is a perfect supervisor and without his encouragement, I am not able to complete my thesis and all projects I did during these two years. He is an expert in computer vision, and is always full of wonderful ideas. Working with him is full of fun. I also want to thank my defence committee members: Dr. Ze-Nian Li, Dr. Mark Drew and Dr. Richard Vaughan, not only for their time but also their patience, helpful suggestions and comments.

I also want to thank all members in the Vision and Media Lab. They are very nice and I have wonderful time being together with them. Thanks Lei Chen for his patience and support. Thanks Wang Yan for giving me lots of guidance on many papers. Thanks Greg's wife for bringing tasty desserts. Thanks Jinling Li for lots help and guidance. Thanks all my collaborators: Lei Chen, Zhiwei Deng, Jinling Li, Mehran Khodabandeh and Srikanth Muralidharan for their hard works.

Last but not least I want to thank my parents. Thanks for always being supportive. Without their support, I will not be here as a student of SFU. And most importantly, thank you for loving me so much. I love you too.

Table of Contents

List of Figures

Chapter 1

Introduction

Object detection is a crucial step in computer vision with wide applications such as visual surveillance, driver-assistance systems and image retrieval. High-level human activity analysis typically builds upon this step. The goal of object detection is to localize an object of certain class such as human, cat or bottle according to the needs of designers of detection system. To localize an object, we can either use a bounding box which can be rectangular, oval, etc. to enclose the targeted objects or assign each pixel a label indicating whether it belongs to an object or not. In order to decide whether the detected objects belong to the classes defined by the designers of detection system, object classification is also a crucial step following object detection. The goal of object classification is to differentiate a given set of objects into several categories while the number of categories is known ahead.

1.1 Motivation

Detection is a difficult problem due to changes of appearances and scales. For example, there are various poses and different shapes of bottles, and objects in one frame have very different scales because of their distances to the camera. In this thesis, we are especially interested in surveillance types of videos. In this setting, the view of the camera is fixed and objects of interests are recorded.

Standard approaches to the problem include background subtraction, moving point trajectory analysis, and appearance-based methods. All of these methods have long histories in the computer vision literature. Appearance-based methods are exemplified by the histogram of oriented gradients (HOG) detector [10] and its variants. Background subtraction-based methods (e.g. [52]) find contiguous foreground regions and further classify them into object categories. Point trajectory [7] or moving region [8] methods are related, finding groups of points moving together or containing a motion different from the background.

Each of these methods has its shortcomings. Appearance-based methods are sensitive to highly textured background regions and need to handle substantial intra-class variation. Background subtraction methods and moving point/region methods are sensitive to moving

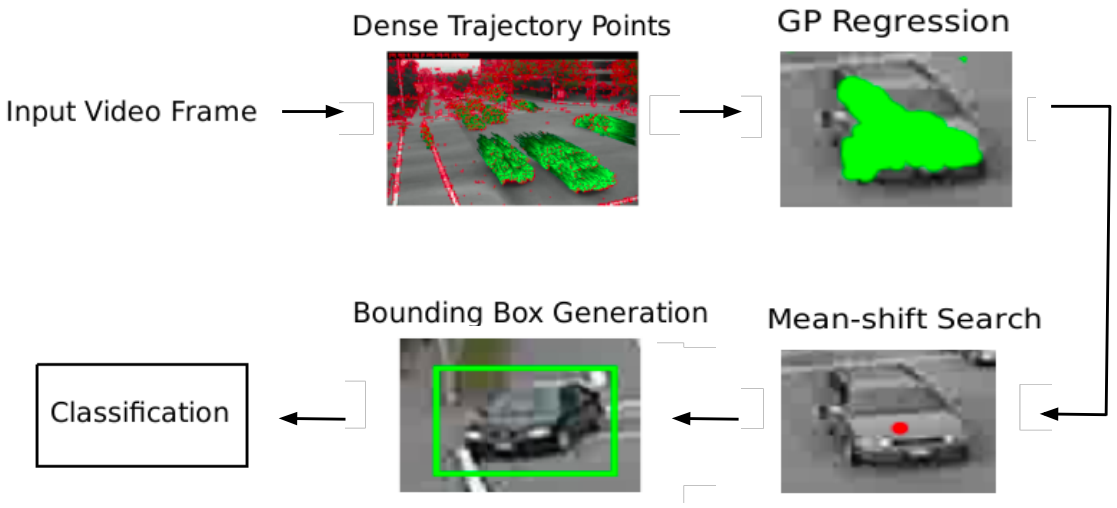

Figure 1.1: Overview of detection procedure. Our method first computes dense trajectories from an input video. Each dense trajectory votes for an object center via a learned regression model. These votes are aggregated using mean-shift clustering. Finally, bounding boxes are generated from the clusters and then scored using a classifier on dense trajectory features.

clutter regions. Further, there is often ambiguity when distinguishing foreground from background due to pixel-wise similarities between objects and the background.

In this paper we present an application of the recently developed dense trajectories approach [50] for combining both moving trajectories and discriminative (moving) appearance-based classification for object detection in surveillance video. The dense trajectories approach has been shown to obtain state-of-the-art performance on standard benchmarks for human activity recognition, particularly for unconstrained internet videos. Here we demonstrate their effectiveness for the tasks of human and vehicle detection in surveillance videos. We focus on detecting only moving objects – note that while stationary objects can be of interest, in a surveillance context objects generally move at some point, and a tracker or scene entry/exit point knowledge can be used to fill temporal gaps.

1.2 Contribution

The contribution of this paper is the application of dense trajectory descriptors to the problem of object detection in surveillance video. Our method utilizes dense trajectories in two steps. First, we detect moving regions and estimate object locations by regressing from dense trajectory descriptors to object locations. After forming candidate detections, we then score them by training a classifier upon a dense trajectory bag-of-words representation. We demonstrate empirically that this method can be effective for human and vehicle detection in surveillance video.

Further, most of the methods try to find the targeted objects directly, while ours tries to firstly find the centers and then given the centers localize objects secondly. The advantages of doing this are that: (1) if we know the object centers, we know roughly how many objects are there in one frame, it is extra information we know ahead before we localize objects; (2) the centers already give us the rough locations of objects.

Difficulties in these tasks include: (1) scale changes of objects are very large, objects with 10 or 150 pixels diagonal are both our targets, (2) the resolution is relatively low, especially for objects which are very small, (3) shadows.

The application of object detection is published in [54]. Further, I also participated in another project [14] which proposed a deep neural-network-based hierarchical graphical model for individual and group activity recognition in surveillance scenes. My roles in these publications are:

- In [54], I implemented the detection system with dense trajectory computation, feature extraction, learning with SVM, etc. And I measured the performance of our system with a series of experiments on two datasets.
- In [14], I participated in the model design and I implemented part of the model about modelling interactions between people in convolutional neural network.

1.3 Overview of the Thesis

In this thesis, we propose an algorithm for the application of object detection. The rest of the thesis is organized as follows. Chapter 2 illustrates the related previous work in object detection with computer vision techniques. Chapter 3 introduces the detection system implementation. Chapter 4 shows the experiment results with two detection tasks on two different datasets and evaluates the performances of detections respectively. Chapter 5 concludes the thesis and provides possible future work.

Chapter 2

Previous Work

As noted above, object detection in surveillance video is a well-studied problem. Turaga et al. [47] provides a survey of this literature. Classic approaches mentioned above include appearance histogram-based methods [10]. More recent methods based on refined Haar-like features [55] and deep learning [32] have shown impressive results for single image person detection. In this thesis, we differentiate detection approaches into the following three categories:

(1) Detect objects as the whole. This means the objects are detected using all features extracted from whole body of object. No configurations between object parts or pixels are learned during the process.

(2) Detection based on parts. This means besides features extracted from whole body of objects, features extracted from finer parts defined as body parts as also used. And the model would also probably model the joint distribution of the body parts.

(3) Detection based on pixels. Features extracted on pixels or from a small patch centered on pixels are used to detect and recognize objects.

2.1 Detect Object as the Whole

Oren et al. proposed an object detection algorithm [36]. The algorithm uses Haar wavelet template to get low-level image features. Given a fixed size bounding box in the training set, the bound boxes are gridded and differential operator is carried inside each grid. SVM is trained on wavelet coefficients to select objects of interests. Gavrila et al. proposed an algorithm [19] in which a template hierarchy which are object shapes are defined. This hierarchy can be built off-line using k-means like algorithm in a bottom-up fashion. And during online process, distance transform based matching scheme is used to match one given image to the pre-defined set of object shapes.

Another fundamental approach falling in this category is the so-called "sliding window approach" as used by Dalal et al. [10]. Given the targeted object classes, classifiers are

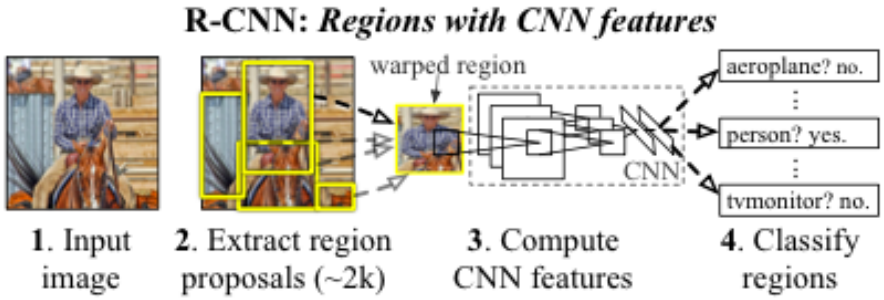

Figure 2.1: RCNN[1]

trained and then a sliding window would scan the whole image across different scale levels, the feature inside the window each time is extracted and the trained classifiers receive these features as inputs, then the score of corresponding window is achieved. Given scores of all windows, non-maximum suppression is adopted to filter out redundant responses on same single person at slightly shifted positions and similar scales. This approach acquires relatively large number of training examples and the detection results rely highly on the quality of training examples and the trained classifiers. Also, this approach generates many more negative examples than positive examples.

A higher level version of "sliding window approach" is a deep learning approach called RCNN proposed by Girshick et al. [20]. They used a method called selective search from [48] to generate region proposals which most likely contain meaningful objects. Selective search generates roughly few thousand image patches which are many more less than proposals generated by a sliding window. Then all the proposals generated would go through a convolutional neural network (CNN) which contains 5 convolutional layers plus two inner product layers. Given classifiers trained on features extracted from CNN, finally scores of all proposals are achieved. This method showed impressive results on object detection and thanks to transfer learning, the model does not need very large number of training examples.

The approach proposed by Blaschko et al. [3] treated the sliding window problem differently: instead of modeling it as a classification problem, they tried to predict structured data, meaning instead of predict binary labels, they predicted the bounding boxes located in the image. This approach is better than standard sliding window approach because for standard sliding window approach, if we want good detection results, all examples shown in test set should also be collected in the training set. If this does not happen, the classification results would be poor. While for this approach, structured training can handle partial detections since loss can be scaled flexibly.

[1]This figure is reproduced from [20]. © 2014 IEEE.

Rujikietgumjorn et al. proposed a detection algorithm [43] based on silhouette shapes. Given pre-computed silhouette shapes of persons from background subtraction or motion analysis, shape covering of foreground mask data is used to generate detection candidates. Unary scores indicating single detections and pairwise scores indicating pairs of occluded detections are defined. The goal is to find a subset of detections which finds the best tradeoff between unary scores and pairwise scores. This method improves sliding window approaches that use non-maximum suppression: partial detections in training set are assigned either positive or negative, this is very bad in the sense that truncations, occlusions and bad localizations happen while the labels assigned to the objects are different. Simple classifiers cannot handle this problem properly while the objective function in this approach can handle it.

2.2 Detection Based on Parts

Based on the careful design of region proposal algorithms and training algorithms, holistic methods can give good results, they are still relatively weak on handling two major problems:

(1) Deformation. Shapes of certain class of objects change a lot: people have different poses such as walking, standing and falling, and the appearances of these different poses change a lot. A simple classifier that treats detection as the whole cannot solve deformation problem intrinsically.

(2) Occlusion. Certain parts of objects may be invisible because they are occluded by another type or same type of objects. Moreover, it is hard to know which or how many parts are invisible.

Compared with object detection as the whole, detection based on parts has more advantages facing these two problems because sometimes existence of distinguishing parts denotes the existence of objects. Approach proposed by Agarwal et al. [1] learns a large vocabulary of object parts and each image is represented by a binary feature vector, each value denotes whether certain part is represented in the image or not. And a classifier is trained on the binary feature vectors to learn the spatial configurations of these object parts. Approach proposed by Burl et al. [6] learns the probability of spatial configurations of joint occurrences of object parts which are selected by hands.

To solve deformation problem, deformable part models (e.g. [17] [18]) have been widely used. These models fall into "detection based on parts" because they model the configurations of fixed number of parts to allow for different appearances of objects.

In the state-of-the-art approach proposed by Felzenszwalb et al. [18], a model is defined with a root filter which models the appearance of the entire object and fixed number of part filters which model the deformable parts in the higher resolution level. The model models both the appearance similarity as the whole and the deformation costs for placements of

different parts. The model works in a sliding window style and the root filter is just as it is in Dalal's paper [10]. However, the final score is sum of root filter score, part filter score minus deformation costs. And the score is achieved by a dot product of weights and HOG features extracted inside windows, either root filter windows or part filter windows.

To model the deformation, approach proposed by Felzenszwalb et al. [18] uses a star model, while approach proposed by Zhu et al. [57] uses a tree model. Different from two-layer DPM model, this model can process both two and three-layer configurations: an object is the mixture of hierarchical tree structured layers and the nodes in each layer represents the parts. Approach in [18] requires carefully selection of parts while this one does not. If there are 9 nodes, namely 9 parts, in the second layer, each part will represent one ninth of the entire object. In the third layer, the nodes part the entire object into finer grids. For example, if there are 36 nodes, the entire object will be grid into 6 by 6 grids and each node will represent one grid.

Classic approaches mentioned above all model parts having semantic meanings: parts are head, torso, legs, etc, which all have anatomical meanings. The approach proposed by Wang et al. [51] instead models the parts to be either rigid parts just as approaches mentioned above or poselets which can cover combinations of parts of the body.

Visibility estimation is a key problem which lots of scholars try to solve. Various approaches were opposed to estimate the part visibilities. An common approach is to run part detectors all over the images and estimate the visibility by hard thresholding the scores returned by the part detector.

Approach proposed by Dai et al. [9] trained substructure detectors and a positive response is achieved only when all related detectors give positive responses. After responses of substructure detectors are gathered, the object detections based on ensemble of part detections are modeled using Markov random field and solved using belief propagation.

In [53] from Wu et al., edgelet features are used for training part detectors. First of all, each edgelet feature is used to train a weak part detector, and a strong detector is a linear combination of weak classifiers. And different from the papers mentioned above, the part detector is a mixture model which models multiple views of each parts. Given scores returned by part detector, visibilities of parts are estimated by hard-thresholding the scores to be larger than a pre-defined value. And the final human locations are estimated by calculating a bayesian combination of visible parts.

The approach above made the assumption that visibility of one part is independent of other part detectors. However, it is not often the case. For example, when the eyes are occluded by the hands, it is not appropriate to say that the head is visible. To take the relations of visibilities of parts into account, Duan et al. proposed a detection algorithm [15]. In this approach, the visibilities of parts are achieved same as above by hard thresholding the detection scores of parts. Then given the rules defining the relations between visibilities of

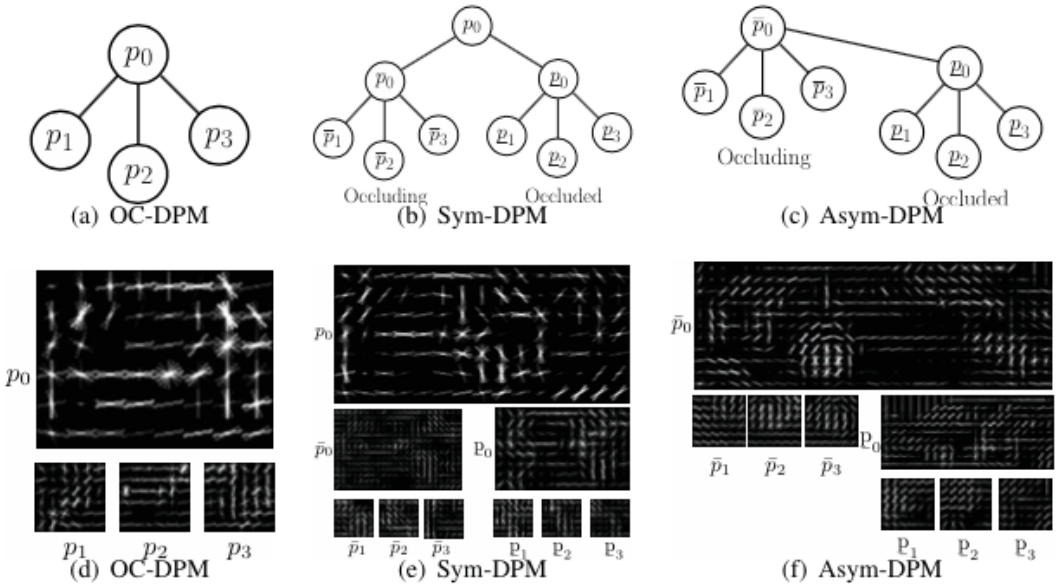

Figure 2.2: Occlusion-based DPM[2]

parts which are manually defined, the compatibility of visibilities of two parts are evaluated by the rules. The detection is positive when the compatibility satisfies the demands.

Hard-thresholding detection scores is not robust enough. Wanli et al. proposed a discriminative deep model [37] [39] for object detection. In the approach, given the scores of part detectors, the visibilities of parts are estimated using a CRF model. The part model consists of three layers of which bottom layer contains parts of smallest size and top layer contains possible occlusion status. Each status in higher level is achieved by either directly passing one status from lower level or passing combinations of status from low level. In this way, the relations of visibilities of parts are built through shared parents instead of modeling the relations directly in the same layer.

It should be also noted that some approaches choose to solve deformation and occlusions simultaneously. For example, in the approach proposed by Pepikj et al. [41] an occlusion-based DPM model is proposed. Instead of modeling single object in the root filter, the model could also model two objects that occlude each other: the two deformable parts at level 1 are two separate objects, deformable parts at level 2 are parts of the targeted objects. One component of this mixture component model could be standard component of DPM which models one view of object, or could be component which models two objects as the whole. The visualization of this model can be found in Fig. 2.2.

[2]This figure is reproduced from [41]. © 2013 IEEE.

2.3 Detection Based on Pixels

For the papers mentioned above, the detection algorithm works by extracting features from a window and using a classifier to classify whether it's a positive patch or negative patch or localizing object parts and modeling the configurations of these parts. However, sometimes people want more precise detection results. Instead of deciding whether an image patch is positive or not, we want to decide whether an image pixel is positive or not: if it's positive, then the pixel belongs to figure; otherwise, it belongs to ground.

Background subtraction [33] is widely used in many computer vision applications. Normally people calculate the background image by calculating the mode value of each pixel given all frames of one video sequence and the foreground images are achieved by subtracting the background image from all frames. And another version [23] is to dynamically calculate the background images while processing one video sequence. Bregler et al. [5] also takes advantage of pixel information. By using optical flow, each pixel is tracked and the motion pattern is recorded. Pixels are clustered using EM clustering algorithm and blobs of pixels with same motion patterns are found.

Leibe published a series of papers on object detection (e.g. [29] [31] [30]). An effective method called "implicit shape model" was proposed in [29]. In the paper, for a given object category, an appearance codebook is built. The codebook stores representative local appearances and the corresponding relative location distributions of the appearances showing on the objects. In other words, the entries in the codebook are centers of small clusters of appearance features which are similar to each other. Given an unseen image, an interesting points detector is run and small image patches are cropped around these detected interesting points. In the next step, the extracted image patches are matched to the entries to the codebook. It should be noted that instead of activating the entry which best matches the patch, all entries are activated if the similarity measures are above a hard threshold. For each activated codebook entry, the corresponding positions relative to object centers are stored and votes for possible centers are gathered. Then a probability density estimation called "parzen window" is used for predicting object locations. And because the interesting points are very sparse, back projection is used to activate image patches which vote for the corresponding object locations and the activated image patches are boundaries of object candidates. The detection procedure is shown in Fig. 2.3. Then given each pixel in activated patches, the possibilities whether it's figure or ground are calculated.

In the approach proposed by Leibe et al. [31], after the initial recognition approach including implicit shape model, segmentation and overlap objects verifications, the authors tried to combine global cues with local cues. Firstly, given detections returned by initial approach, chamfer matching is used to refine object segmentation consider global constraints. Chamfer match refers to the procedure that given a set of templates, locations where these templates are matched better are searched and returned. In this paper, the templates are

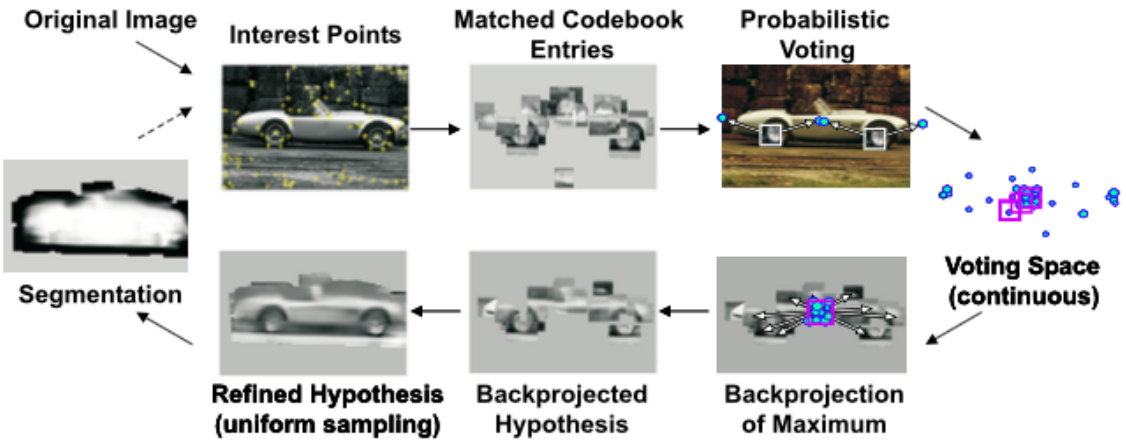

Figure 2.3: Implicit Shape Model[3]

object silhouettes. Moreover, MDL verification is used to allow for part instead of full segmentation consistency with silhouettes.

Angelova et al. proposed a segmentation algorithm in [2]. In the paper, firstly a region based detector is run to get regions that possibly contain targeted objects. And then Laplacian-based propagation is used to do a full segmentation inside positive regions. Gould et al. proposed another region based segmentation algorithm [21] which instead of reasoning regions and pixels separately, reasons about pixels, regions and objects simultaneously. For a given pixel in the image, it is assigned specific class label of background classes or foreground class ignoring specific class at region level. Then at object level, the foreground classes of pixels are decided. What's more, the model can process objects consist of multiple regions and reasons about the joint movement of these regions in object level. Furthermore, the model considers context information of regions. For example, "air plane" is flying in the "sky". An overview of the detection system is shown in Fig. 2.4.

2.4 New Trend: Detection under Deep Learning Framework

Recently, deep learning has gained so many attentions. A deep neural network is normally a feed forward neural networks with many hidden layers, and is always trained with back propagation algorithm. "Deep learning" is named to differentiate previous shallow structured neural networks. For example, ELM [24] [25] proposed by Huang et al. is a single hidden layer neural network. Instead of training using back propagation , it is trained by computing a generalized inverse which is also named Moore-Penrose pseudo-inverse. Compared to shallow structured neural networks, deep neural networks have better generalization ability

[3]This figure is reproduced from [29].

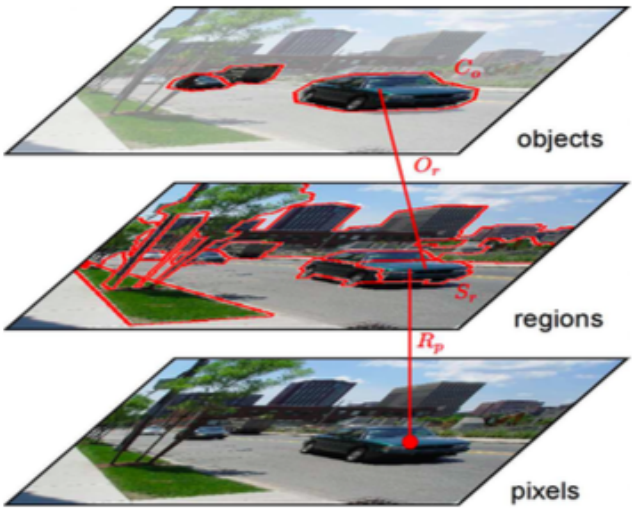

Figure 2.4: Region based detection algorithm proposed by Stephen Gould et al.[4]

and better performances. As reported in [20], RCNN boosts the mean average precision by 30% compared to previous best detection results on VOC 2012.

With the success of RCNN, some scholars show great interests in this piece of work and improvements on RCNN are proposed. Girshick et al. has pointed out in the error analysis part that most errors are caused by bad localizations. Zhang et al. proposed an algorithm [56] to address the localization difficulty that RCNN faces. In the paper, they developed an algorithm which can generate new bounding boxes which are expected to have higher detection scores than previous bounding boxes generated by selective search. Also, instead of training CNN using softmax loss layer, the CNN is trained using structured loss: a structured SVM is implemented in the loss layer targeting more precise localizations. The detection mAP achieves 66.4% on VOC 2012 which shows 10% higher accuracy than RCNN (mAP 53.3%), and it should be noted the IoU overlap > 0.7 is used for purpose of precise localization.

It is commonly known that training a deep neural network successfully demands huge amount of image patches. For example, Krizhevsky et al. [27] shows substantially higher image classification accuracy on the ImageNet [13] [12]. Their success mostly resulted from training a model on very large dataset with 1.2 million labeled instances. Recently, Oquab et al. proposed a weakly supervised deep learning framework [35] for object localization and object recognition. In the paper, image level labels are provided but not the locations of the objects. All image patches are resized to scales within certain ranges which are pre-defined and then image patches are cropped in a sliding window manner. Global max pooling

[4]This figure is reproduced from [21].

is used across all windows in a given image of certain category to activate the windows with highest score. The objective function of loss layer is modified to allow for multi-class classification. Though the algorithm is weakly supervised but it still shows great potentials in localizing objects. However, the algorithm can only locate one object given one object category.

Traditional convolutional neural networks output the posterior probabilities of a given image patch. Tompson et al. proposed a CNN architecture [45] [46] which models the joint distribution of body parts to refine detection proposals. The body parts are detected in a sliding window manner and the outputs are heat-maps which model the spatial locations of body parts. Given the detected body parts which provide unary scores, the pairwise scores indicating conditional distribution of one body part given another are modeled by connecting each pair of body parts thus resulting in a fully connected architecture. The spatial model can be seen as one round belief propagation implemented under convolutional neural network framework and if trained with stochastic gradient descent, can be jointly trained together with single part detector mentioned before. The CNN implementation of this model is shown in Fig. 2.5.

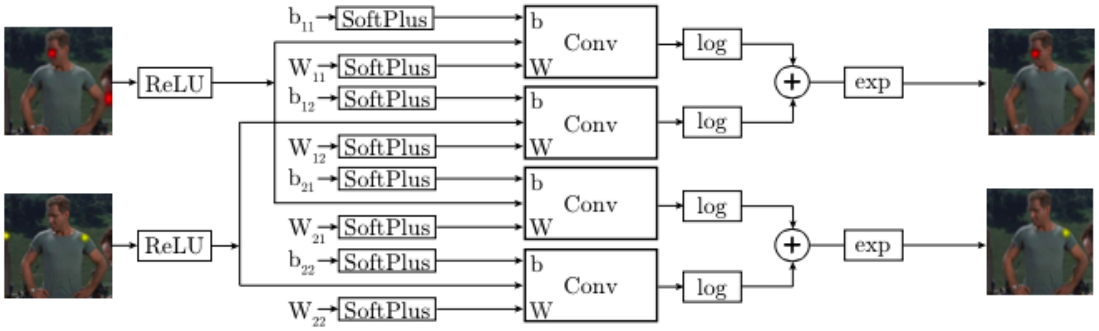

Figure 2.5: CNN structure proposed by Tompson et al.[5]

Just like what scholars tried to solve before, precise localization such as segmentation is a key target. Recently, Hariharan et al. [22] proposed a CNN structure targeting object segmentation. To allow for precise localization, first concern is that features extracted from last layer may be too coarse, stacking features from multiple layers may be a better choice since convolution layers gather more detailed information of the objects. Secondly, pixel classification is needed to achieve object segmentation. They conduct pixel classification by training fine-grained classifiers and the score of each pixel is calculated by summing up scores returned by classifiers which are within neighboring region. Also the classifiers are location specific to take part locations into considerations.

[5] This figure is reproduced from [46].

A recent work [40] improves accuracy and efficiency of deep convolutional neural networks (DCNNs) by introducing epitomic image representation [26] into neural networks. A DCNN detector works in the sliding window manner which uses epitomic convolution instead of normal max-pooling. A deformable deep convolutional neural networks for object detection [38] is proposed. A def-pooling layer is defined which calculates pooling within part blocks instead of calculating global max-pooling.

In this thesis we focus on surveillance video. Analyzing the temporal domain should lead to more robust detection algorithms for this static camera setting. Classic methods such as grouping moving feature points exist [7]. Similar methods have been explored in the context of crowded videos of people [42]. The aforementioned appearance-based methods have been extended to video, e.g. [11, 49]. However, relatively fewer recent methods that focus on motion patterns exist. The main focus of this thesis is revisiting the idea of trajectory-based object detection and classification based on state-of-the-art trajectory descriptors.

Chapter 3

Motion-based Detection Model

A detailed overview of our method is shown in Fig. 3.1 and the whole algorithm we used to detect objects is shown in Algorithm 1. First, we follow the dense trajectory pipeline of finding and tracking moving points over short time scales. From these we use regression to predict the positions of objects. We then agglomerate nearby predictions by clustering and generate bounding boxes. After several optional verification steps which further refine the detections, finally classifiers are trained to discriminate objects of interest from other regions. In the following subsections we provide details of each of these steps.

3.1 Trajectory-aligned Motion Features

We develop an algorithm for detecting moving objects in surveillance video. The first step of our algorithm is to generate a set of candidate object locations. Analagous to Hough transform-type voting methods (e.g. [28]) we will do this by generating an initial set of points that can vote for possible object centers via a regression step. We use the dense trajectory algorithm from Wang et al. [50]. In that algorithm, dense trajectories are obtained by tracking densely sampled image points through multiple frames using optical flow, HOG descriptors are extracted around each point along a trajectory and normalized to produce a feature vector for corresponding trajectories. An example frame visualizing dense trajectory is shown in Fig. 3.2.

We chose dense trajectory features because they have certain advantages over other features.

1. A key consideration is that we would like a large number of such points. In contrast, approaches such as Harris interest points are sparse and will have difficulty covering objects, especially given noise in the subsequent regression step. Our approach is to extract dense moving points from a given video, and use these points to vote for possible object centers. And these points are used to generate a region later which possibly covers an object.

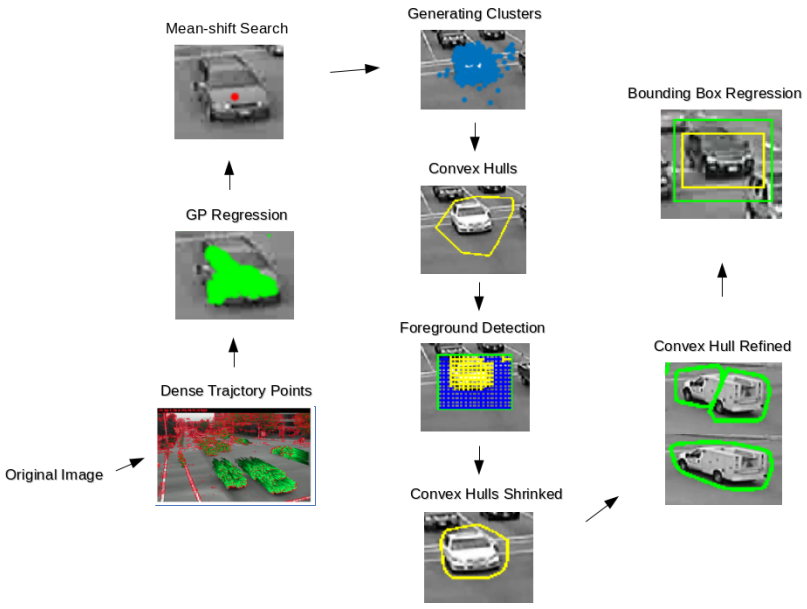

Figure 3.1: Detailed system overview

2. The dense trajectory algorithm abandons the idea of tracking object as the whole. Thus it is more robust to partial occlusions. Even when the object is partially occluded, it is still possible to track this object, which means even when the center of object is occluded, it is still possible to regress its centers given tracked points on this object.

3. Features extracted from trajectories are shown to be more robust compared to features extracted from single points. Given results from paper [44], dense trajectory features are definitely among top hand-crafted features.

We use these densely sampled trajectory points and their descriptors as inputs to regress object positions, as described next.

3.2 Center Prediction Based on Regression

3.2.1 Regress the Centers

In this section, we describe how possible object center locations are predicted using a regression model and how to detect objects given these predicted centers. An example frame is shown in Fig. 3.3.

Let the number of trajectories be N and the length of each trajectory be L. Our goal is to predict the locations of object centers, given a feature vector s_i extracted along a

Data: target videos

Result: rectangular bounding boxes and corresponding scores and labels

1 Run dense trajectory algorithm [50] to get dense trajectory points P and features Φ ;

2 Run multi-output Gaussian process regression algorithm [4] to regress possible centers C^1 ;

3 Manually define near-field and far-field regions, run mean-shift clustering algorithm separately in world coordinate in near-field region and image coordinate in far-field region to find maxima of center locations C^2 ;

4 Project C^2 into world coordinate, use Algorithm 2 to refine C^2 and get refined centers C^3, then project C^3 back to image coordinate to get final centers C ;

5 Generate clusters of points given C;

6 Compute convex hulls and tight bounding boxes given all clusters of points ;

7 Shrink convex hulls by gridding the bounding boxes and pruning cells with too few dense trajectory points inside ;

8 [optional] Group neighboring bounding boxes with shorter lengths ;

9 [optional] Split one bounding box into two bounding boxes with different moving directions ;

10 [optional] Regress a new bounding box for each bounding box in far-field region ;

11 Train a classifier to differentiate objects with backgrounds ;

Algorithm 1: Detection algorithm

trajectory where $\{i = 1, 2, ..., N\}$. A regression model is learned and the output of the regression model is the offset vector $o_i = (x_i, y_i)$ starting from points $p_{ij} = (x_{ij}, y_{ij})$ on a trajectory pointing to possible centers $c_{ij} = (u_{ij}, v_{ij})$ where $\{j = 1, 2, ..., L\}$. Because there is only one feature vector extracted from the trajectory but there're l points on the trajectory, we assume all points on a trajectory should share exactly the same offset vector. Thus, the inputs of the regression model are features s_i and the outputs are o_i. The center prediction can be computed as:

$$c_{ij} = p_{ij} + o_i \tag{3.1}$$

Many objects have internal symmetries. For example, a car has two front lights. This requires the model to have the capability that given one feature vector the output should not be only one offset vector but several possible offset vectors. Suppose we want to produce M outputs given one input. Intuitively we can achieve this goal by clustering the input feature space into M subspaces, train a regression model for each subspace, and finally get M outputs given one input. The problem of doing so is that we cannot capture the relations between pairs of outputs. Sometimes more accurate results can be generated if the relations between outputs are also modeled. To achieve this goal, we use the dependent

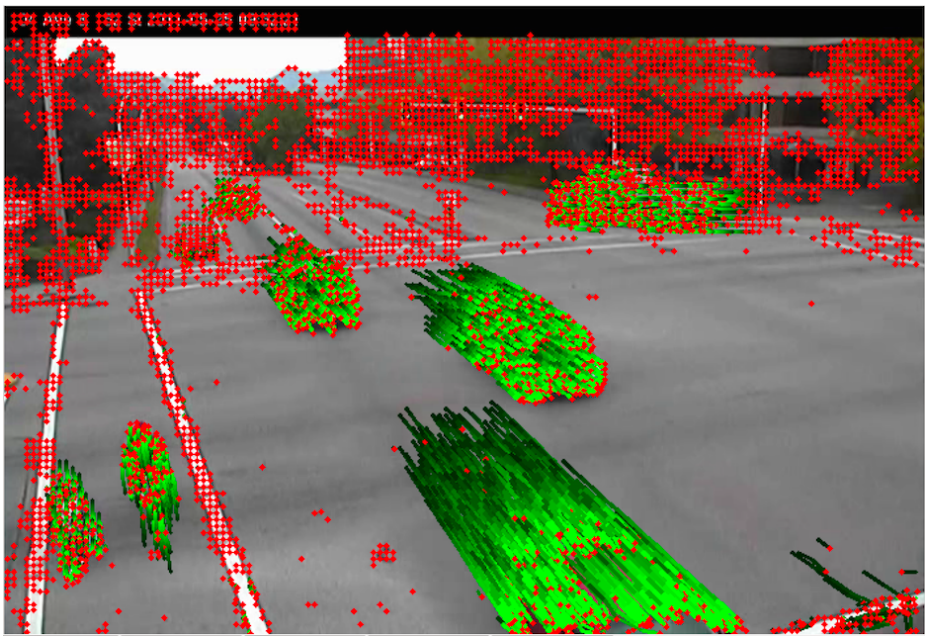

Figure 3.2: Example Frame of Dense Trajectory

gaussian process model [4] where the relations between the outputs are modelled by adding a noise source which influences all outputs.

Suppose we want to produce M offset outputs $Y_k(s)$ where $s \in R^p$ is a dense trajectory feature and $k = \{1, 2, ..., M\}$, and for each output we have N_k training observations. Given M datasets $D_k = \{s_{ki}, o_{ki}\}_{i=1}^{M_k}$, we want to learn a model from the combined data $D = \{D_1, D_2, ..., D_M\}$ to predict $(Y_1(s'), Y_2(s'), ..., Y_M(s')$ for input $s' \in R^p$. Each output is modeled as a sum of three gaussian processes U, V, and W. V is unique to each output, U shares the same noise source to ensure the outputs are not independent, and W is additive noise. Thus we have $Y_k(s) = U_k(s) + V_k(s) + W_k(s)$. Let the covariance matrix be Cov^Y, then we have $Cov^Y = Cov^U + Cov^V + \sigma^2$. Following [4]:

$$Cov^U_{kk}(d) = \frac{\pi^{\frac{p}{2}} r_k^2}{\sqrt{|A_k|}} exp(-\frac{1}{4} d^T A_k d) \tag{3.2}$$

$$Cov^V_{kk}(d) = \frac{2\pi^{\frac{p}{2}} w_k^2}{\sqrt{|B_k|}} exp(-\frac{1}{4} d^T B_k d) \tag{3.3}$$

$$Cov^U_{kk'}(c) = \frac{\pi^{\frac{p}{2}} r_k r_{k'}}{\sqrt{|A_k + A_{k'}|}} exp(-\frac{1}{2} d^T A_k (A_k + A_{k'})^{-1} A_{k'} d) \tag{3.4}$$

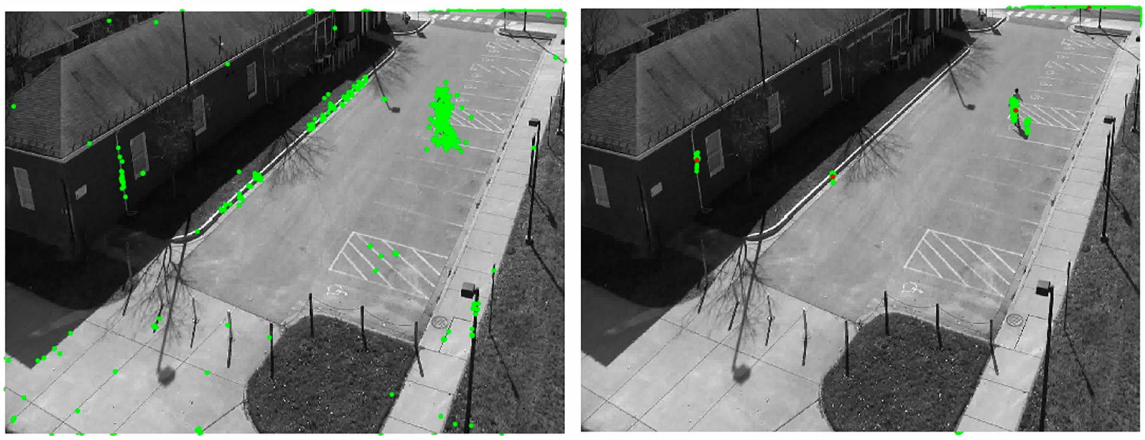

Figure 3.3: Example Frame of Regressing the Centers: green points in left frame are raw dense trajectory points in this frame; green points in right frame are centers which are outputs of regression algorithm and red point is the output of mean-shift clustering.

Where r, w, A, and B are parameters of gaussian kernels, and d are separation between two inputs s_i and $s_{i'}$. Given $Cov^Y_{kk'}$, the covariance matrix C is constructed as below:

$$\begin{bmatrix} Cov^Y_{11} & \dots & Cov^Y_{1M} \\ \vdots & \ddots & \vdots \\ Cov^Y_{M1} & \dots & Cov^Y_{MM} \end{bmatrix} \tag{3.5}$$

Now we can compute the log-likelihood:

$$L = -\frac{1}{2}log|C| - \frac{1}{2}\vec{o}^T C^{-1}\vec{o} - \frac{N}{2}log(2\pi) \tag{3.6}$$

where C is a function of parameters $\{r, w, A, B, \sigma\}$. And the mean μ of gaussian kernel is set to $\vec{0}$ in our algorithm. Learning a model corresponds to maximizing log-likelihood L, in our algorithm, the parameters are learned with gradient descent. The predictive distribution at the k^{th} output is a gaussian with mean $\hat{\mu} = \vec{q}^T C^{-1}\vec{o}$ and variance $\hat{\sigma} = \kappa - \vec{q}^T C^{-1}\vec{q}$, where $\kappa = Cov^Y_{kk}(0)$ and $\vec{q} = \{C^Y_{k1}(s' - s_{11}), ..., C^Y_{k1}(s' - s_{1N_1}), ..., C^Y_{kM}(s' - s_{N_M 1}), ..., C^Y_{kM}(s' - s_{N_M N_M})\}$.

Using the algorithm above, each trajectory will propose M possible center locations in one frame. Given a set of predicted center locations we want to find maxima in the continuous center space, thus mean-shift clustering is used to generate object hypotheses. Firstly, mean-shift clustering can be used in either image coordinates or world coordinates (if camera calibration information is available). Secondly, if object scales change drastically, mean-shift clustering with different bandwidths can be used for different regions. An ex-

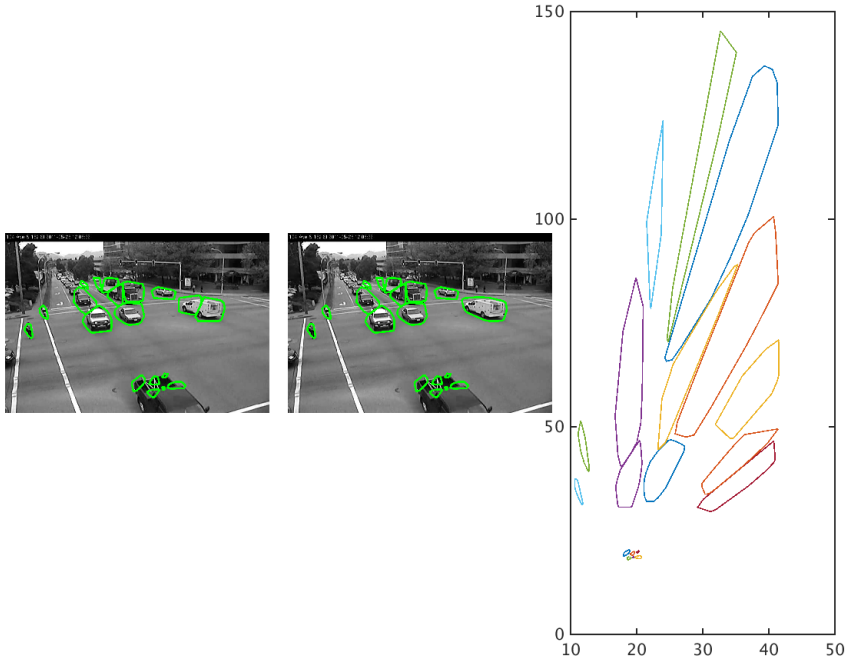

Figure 3.4: Bounding boxes in image and world coordinates: right figure shows the bounding boxes in the world coordinates, which see every object from its top; other two figures show the bounding boxes in the image coordinates, which see every object from the view of the camera.

ample frame showing bounding boxes in image coordinates and world coordinates is shown in Fig. 3.4.

Ideally objects in the world coordinates should have similar sizes. Though objects in near-field are mostly in similar sizes, objects in far-field regions are distorted because that camera calibration information is not perfect. To get more precise information about center locations, we combine objects in world coordinates in near-field regions with objects in image coordinates in far-field regions when running mean-shift clustering algorithm.

3.2.2 Refine the Centers

The results returned by mean-shift clustering in near-field regions are not perfect because of the sparse dense trajectories and large scale. To refine the results, Algorithm 2 is used. The inputs to the algorithm are outputs of mean-shift algorithm in world coordinate and the outputs of this algorithm are the refined centers locations. Different from mean-shift algorithm which tries to find cluster of points near center of the cluster, this algorithm tries to find cluster of points so that for each point inside the cluster at least one other point inside same cluster is near. An example showing the results of this algorithm is shown in Fig. 3.5.

Data: $c_1, c_2, ..., c_n$ which are outputs of mean-shift in world coordinate
Result: refined centers $C = \{c_1^*, c_2^*, ..., c_m^*\}$
Initialization $T = \{c_1, c_2, ..., c_n\}$, $S = \{T_1\}$ where T_1 is the first element in T, $C = \{\}$
while *T is not empty* **do**
  while $t \in T$ *near* $s \in S$ **do**
    $S = S \bigcup t$
  end
  $C = C \bigcup (\frac{\sum(S_x)}{\|S\|}, \frac{\sum(S_y)}{\|S\|})$;
  $T = T - S$;
  $S = \{T_1\}$;
end

Algorithm 2: Center verification after mean-shift clustering

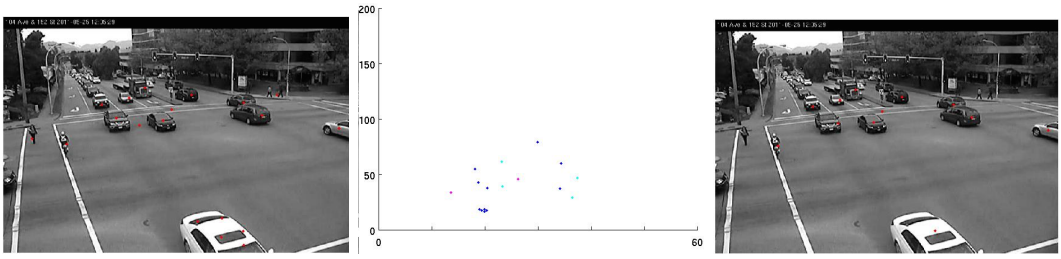

Figure 3.5: Example Frame of Refining the Centers: the left figure shows the center locations after mean-shift clustering, figure in the middle shows the centers in world coordinate and the right figure shows the centers after the verification step.

3.3 Detection Based on Center Prediction

Previous section introduced how to get initial center predictions using regression from dense trajectories. The goal of this section is to get a rectangular bounding box for each object hypothesis given initial set of centers.

1. Given initial set of centers returned by verification step following mean-shift clustering, each moving point is assigned to the nearest cluster center. If there are K centers returned by mean-shift, we will have K object hypothesis each of which is a set of points.

2. Given one object hypothesis, we compute the convex hull of this cluster and this convex hull defines the region covered by this object or the irregular shape boundary of this object.

4. Given a convex hull, we compute a tight bounding box which encloses the object region.

5. As is often the case, objects have shadows and points belonging to shadows also move together with objects. Beyond this, the generation of moving points may be noisy – e.g. a few points far from real moving regions may be generated. Thus the convex hulls may also cover certain areas of the background. To shrink the convex hulls, we firstly grid the bounding boxes and then grid cells are pruned if there are too few moving points inside.

6. Taken the points inside the remaining grids cells, we compute the convex hulls and tight bounding boxes again.

7. Optional verification. Over-segmentation is a common problem in the field of object detection. It means one object is split into two segmentations. To address this problem, we project all convex hulls to the world coordinate, if the length of the convex hulls is too short, we search for nearby convex hulls also with short lengths. we combine the neighboring convex hulls by grouping the points inside two convex hulls together and re-compute the new convex hull. This process is repeated until there is no two convex hulls with short lengths near each other. An example is shown in Fig. 3.6.

8. Optional verification. Under-segmentation is another common problem in the field of object detection. It means one segmentation fully covers two objects. To address this problem, given trajectory points inside each bounding box, we compute the direction (towards camera or leaving camera) in which each point moves. We split one segmentation by grouping together trajectory points with same directions and re-compute the convex hulls. An example is shown in Fig. 3.7.

9. Optional verification. Though we are able to shrink the bounding boxes by pruning the cells with fewer trajectory points, in the area containing dense moving objects, the background still is covered by dense trajectory points. To address this problem, inspired by the bounding box regression strategy proposed in [20], we use a simple but effective regression model to further adjust the positions of the bounding boxes. For surveillance type of video, because the interested targets have fixed types and location of camera is fixed, it is easy to estimate the size of objects appearing at certain location. We learn a regression model which takes the position, width and height of a bounding box, and outputs the predicted width and height of the new bounding box. We assume the center is unchanged.

3.4 Classification

Given all candidate bounding boxes produced by our method, the final step is to score them and decide whether they contain an object of interest. For example, we might be interested

Figure 3.6: Optional verification targeting over segmentation

Figure 3.7: Optional verification targeting under segmentation

in detecting the people in a scene or the vehicles in a scene. We do this by training a binary classifier based on dense trajectory features. The features we used in the previous steps are raw dense trajectory features, while in this section we need to use features which can represent the whole detected objects. An naive approach is to use Bag of Words (BoW) representations of the features which treat visual features as words.

1. Extract features from trajectory points. We assume all points on one trajectory share the same feature vector extracted from this trajectory.

2. Codebook generation. We extract dense trajectory features from the training clip and perform k-means algorithm over all feature vectors, where $k = 100$.

3. Feature representation. We consider all dense trajectory features that pass through a candidate bounding box in one frame. Each bounding box is represented by computing histograms of codewords which are the centers of clusters.

A discriminative classifier (SVM) with linear kernel is trained from a labeled training data set to classify each candidate bounding box as containing an object of interest or not.

Chapter 4

Experiments

We test our model on two object detection problems in surveillance video: vehicle detection and human detection.

4.1 Traffic Dataset

Our model is firstly tested on a traffic dataset, where the focus is on detecting vehicles. Example frames can be seen in Fig. 4.2(top). The training set contains 500 frames of size 480×704 pixels. The test clips contains 2258 frames and are evaluated every 5 frames; ground-truth is obtained via manual labeling.

For training the regression model, all features extracted from the training set are clustered into 10 subsets to generate 10 outputs. 50 trajectories from each cluster that pass through ground truth bounding boxes are randomly picked to form the inputs of the regression model.

In testing, we mark a region of interest corresponding to regions of the video frame where vehicles are of sufficient size. For this dataset, the scale changes are very large: 250 pixels diagonal to 20 pixels diagonal. It is difficult to find a perfect bandwidth for entire images: if the bandwidth is too small, vehicles in larger scale will be over-segmented and if the bandwidth is too large, vehicles in smaller scale will be grouped into one cluster. To address this problem, the image coordinates are manually divided into two regions. For near-field regions, the vehicles have relatively large scale and thus mean-shift clustering is performed in world coordinates with bandwidth 3.5. For far-field regions, mean-shift clustering is performed in image coordinates with bandwidth 20. Note that in a fixed-camera surveillance setting, these parameters could be obtained with camera calibration.

To further improve the quality of the bounding boxes, we refine bounding boxes in two steps.

(1) Bounding boxes refinement. First, the convex hulls of objects are projected into world coordinates. If the lengths of two objects in world coordinates are below a threshold and the centers of these two objects are very close, these convex hulls will be merged

into one convex hull. The information of directions of moving points can also be used to generate finer convex hulls. If there are two objects going in opposite directions, the convex hull will be split into two convex hulls: points inside the convex hull are split into two groups according to their moving direction, the new convex hulls are computed with two groups of points.

(2) Bounding boxes regression. Finally, a simple model is learned to adjust bounding box sizes based on their image coordinates. This is a simple camera calibration-type model, which takes positions of centers of tight bounding boxes of convex hulls as input and outputs the width and height of rectangular bounding boxes consistent with bounding boxes of vehicles at this image position in training data.

Given the dense trajectories points and features, training takes 4 hours to converge and during test time, every 10K trajectories take 23 minutes to process.

We compare every step of our method with a baseline which is background subtraction plus the same classifier used in our approach. We summarize the comparison of our model with the baseline in precision-recall curves shown in Fig. 4.1. The red curve corresponds to our final detection result and the blue curve corresponds to the result of background subtraction. We get roughly 0.85 precision and the precision drops at the end of the curve because a set of false positives which appear at the end of the sorted list of detections get the lowest scores. And we get roughly 0.9 recall since for some objects we are not able to generate detections with more than 0.5 overlap with them.

4.2 VIRAT Dataset

The second dataset we use is the VIRAT dataset [34]. We focus on detecting moving people in the video sequences. The VIRAT dataset contains a large amount of static surveillance camera video. We use Scene 0000, a parking lot scene containing people along moving background clutter such as vehicles.

We define a training set containing 500 frames with frame size 1080×1920 pixels. Again, for regression all features extracted from the training set are clustered into 10 subsets to generate 10 outputs. 50 trajectories from each cluster going through ground truth bounding boxes are randomly picked to form the inputs of the regression model.

Our test clips contains 1900 frames and are evaluated every 2 frames. The image coordinates are manually divided into two regions, mean-shift clustering is performed in these two regions separately. The bandwidth of the near-field and far-field regions are 50 and 25, respectively. Detections with size smaller than a threshold are removed. The two verification steps used in traffic dataset are not used on this dataset. We compare our method to baselines of DPM [16], trained on the same positive data and including hard negative mining, and background subtraction. We consider performance both on all people in the test set ("all people") and only those that are moving ("moving people"). The comparisons

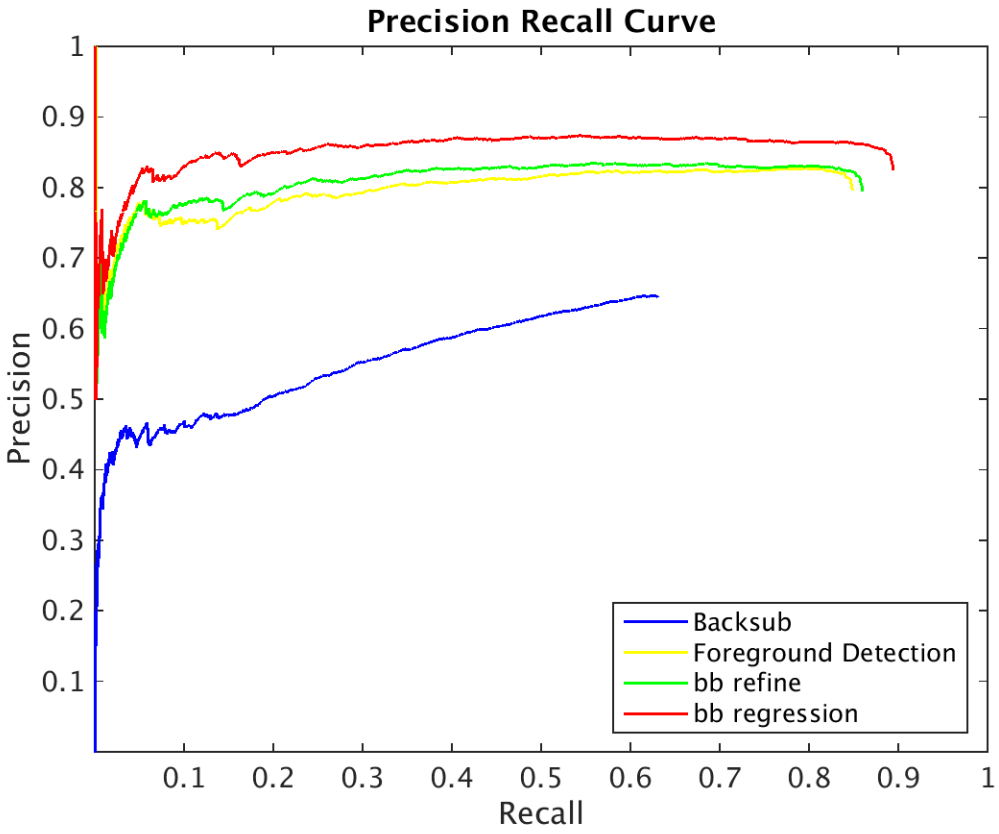

Figure 4.1: Precision Recall Curves.

of final detection results with baselines are shown in Fig. 4.3, using the standard $i/u > 0.5$ criterion. Note that our methods (red and blue curves) achieve higher precision than the baselines. As expected, the static-person DPM detector achieves higher recall for the "all people" setting.

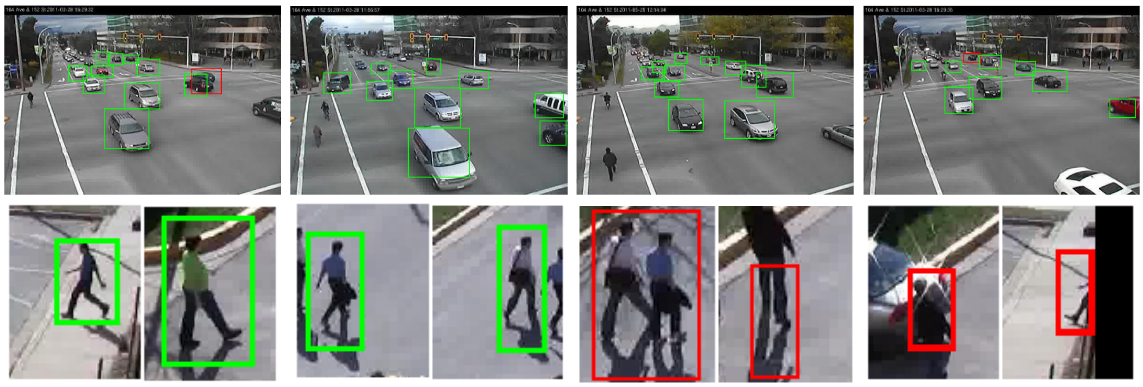

Figure 4.2: Visualization of detection results. Top row shows vehicle detection results on the Traffic dataset, bottom row shows human detection results on the VIRAT dataset. The green bounding boxes are true positives and the red bounding boxes are false positives. For the second row, the first 4 examples are top scoring true positives, the last four examples are the top scoring false positives.

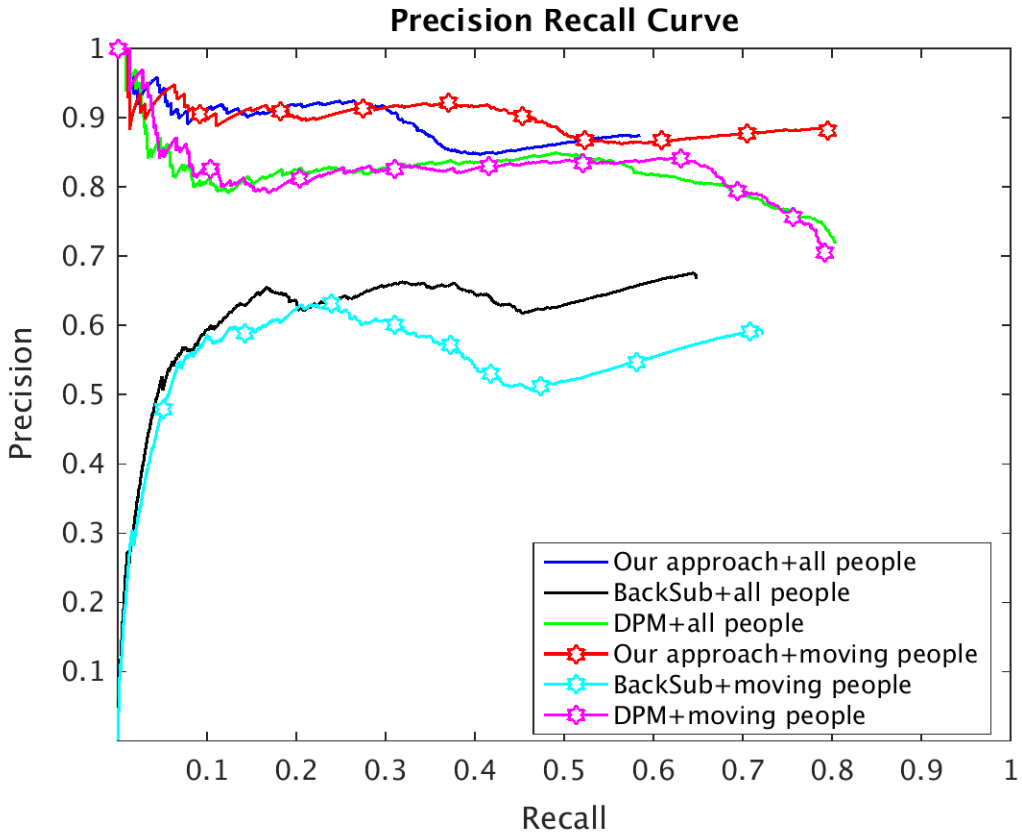

Figure 4.3: Precision Recall Curves. "Moving people" is evaluation only on ground-truth people who are moving, "all people" is the entire set (moving and stationary).

Chapter 5

Conclusion and Future Work

A detection algorithm which focuses on the application of dense trajectories to object detection in surveillance video is proposed. The method shows it is possible to regress object centers and generate rectangular bounding boxes from dense trajectories. We demonstrate our algorithm on two datasets for two tasks: vehicle detection on Traffic Dataset and human detection on VIRAT Dataset. The regression method can generate candidate detection bounding boxes superior to a baseline based on background subtraction. This is possible because regression from moving points to possible centers are advantageous in that moving points densely cover the moving objects. Another factor to the success of our method is that the features extracted from trajectory are quite stable. In summary, our method demonstrates that dense trajectories are effective for object detection in surveillance video.

5.0.1 Limitations

In this thesis, we demonstrate our model on two datasets for two detection tasks: vehicle detection and human detection. The experimental results suggest our model is promising. However, there are certain limitations to our works:

1. If objects belonging to the background shake (e.g., trees), there will be lots of candidates generated corresponding to the shaking background. However, this can be solved by thresholding length of trajectories across multiple frames to be above certain value.

2. It can only detect moving objects. If the object does not move, it cannot be detected because there are no trajectories generated for this object. Also, if the object moves in the way like rotation, it cannot be detected neither. We should also be careful that filtering points which do not move long distance enough across multiple frames may cause miss detection of objects that jump up and down.

3. If the camera shakes, the objects cannot be detected correctly. This is because if the camera shakes, the dense trajectory algorithm will detect all pixels in the frame as

moving points. With the wrong results returned by dense trajectory algorithm, it is not possible to detect objects using our algorithm.

5.0.2 Future Work

The current model does not contain any occlusion handling part, the algorithm accepts every region proposal as a single detection. A direction to go is to add multi-object hypothesis to the classification part and reason about whether the proposal contains single detection or multiple detections. This requires lots of training instances that contain under-segmentations and training instances containing various appearances of under-segmentations may be necessary. And with the popularity of deep learning, it is interesting to know that whether it is possible to incorporate our algorithm into deep learning framework. An easy approach is to extract deep learning features centered around each point using a pre-trained CNN model given the detected dense trajectory points. It is very hard to train a CNN model for our detection system because currently there is no published algorithm which focuses on tracking dense points using deep learning. Further, since our algorithm is based on pixels, it is possible to do object segmentation in deep learning framework by making use of lately published algorithm [22].

Bibliography

[1] Shivani Agarwal and Dan Roth. Learning a sparse representation for object detection. In *Computer Vision-ECCV 2002*, pages 113–127. Springer, 2002.

[2] Anelia Angelova and Shenghuo Zhu. Efficient object detection and segmentation for fine-grained recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 811–818. IEEE, 2013.

[3] Matthew B Blaschko and Christoph H Lampert. Learning to localize objects with structured output regression. In *Computer Vision–ECCV 2008*, pages 2–15. Springer, 2008.

[4] Phillip Boyle and Marcus Frean. Dependent gaussian processes. In *NIPS*, 2005.

[5] Christoph Bregler. Learning and recognizing human dynamics in video sequences. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 568–574. IEEE, 1997.

[6] Michael C Burl, Markus Weber, and Pietro Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Computer Vision-ECCV 1998*, pages 628–641. Springer, 1998.

[7] B. Coifman, D. Beymer, P. Mclauchlan, and J. Malik. A realtime computer vision system for vehicle tracking and traffic surveillance. *Transportation Research C 6C*, 4:271–288, Aug 1998.

[8] Ross Cutler and Larry Davis. Robust real-time periodic motion detection, analysis, and applications. *PAMI*, 22(8), 2000.

[9] Shengyang Dai, Ming Yang, Ying Wu, and Aggelos Katsaggelos. Detector ensemble. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

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

[11] N. Dalal, B. Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

[12] Jia Deng, Alex Berg, Sanjeev Satheesh, H Su, Aditya Khosla, and L Fei-Fei. Imagenet large scale visual recognition competition 2012., 2012.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[14] Zhiwei Deng, Mengyao Zhai, Lei Chen, Yuhao Liu, Srikanth Muralidharan, Mehrsan Javan Roshtkhari, and Greg Mori. Deep structured models for group activity recognition. *arXiv preprint arXiv:1506.04191*, 2015.

[15] Genquan Duan, Haizhou Ai, and Shihong Lao. A structural filter approach to human detection. In *Computer Vision–ECCV 2010*, pages 238–251. Springer, 2010.

[16] Pedro Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), 2010.

[17] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[18] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[19] Dariu M Gavrila and Vasanth Philomin. Real-time object detection for smart vehicles. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 87–93. IEEE, 1999.

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[21] Stephen Gould, Tianshi Gao, and Daphne Koller. Region-based segmentation and object detection. In *Advances in neural information processing systems*, pages 655–663, 2009.

[22] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. *arXiv preprint arXiv:1411.5752*, 2014.

[23] Ismail Haritaoglu, David Harwood, and Larry S Davis. W 4 s: A real-time system for detecting and tracking people in 2 1/2d. In *Computer VisionECCV'98*, pages 877–892. Springer, 1998.

[24] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990. IEEE, 2004.

[25] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[26] Nebojsa Jojic, Brendan J Frey, and Anitha Kannan. Epitomic analysis of appearance and shape. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 34–41. IEEE, 2003.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[28] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, 2005.

[29] Bastian Leibe, Ales Leonardis, and Bernt Schiele. In Jean Ponce, Martial Hebert, Cordelia Schmid, and Andrew Zisserman, editors, *Toward Category-Level Object Recognition*, chapter An Implicit Shape Model for Combined Object Categorization and Segmentation, pages 508–524. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[30] Bastian Leibe, Aleš Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International journal of computer vision*, 77(1-3):259–289, 2008.

[31] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 878–885. IEEE, 2005.

[32] P. Luo, Y. Tian, X. Wang, and X. Tang. Switchable deep network for pedestrian detection. In *CVPR*, 2014.

[33] Atsushi Nakazawa, Hirokazu Kato, and Seiji Inokuchi. Human tracking using distributed vision systems. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 1, pages 593–596. IEEE, 1998.

[34] Sangmin Oh, Anthony Hoogs, Amitha Perera, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011.

[35] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[36] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian detection using wavelet templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 193–199. IEEE, 1997.

[37] Wanli Ouyang and Xiaogang Wang. A discriminative deep model for pedestrian detection with occlusion handling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3258–3265. IEEE, 2012.

[38] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Chen-Change Loy, et al. Deepid-net: Deformable deep convolutional neural networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2015.

[39] Wanli Ouyang, Xingyu Zeng, and Xiaogang Wang. Modeling mutual visibility relationship in pedestrian detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3222–3229. IEEE, 2013.

[40] George Papandreou, Iasonas Kokkinos, and Pierre-André Savalle. Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 390–399, 2015.

[41] Bojan Pepikj, Michael Stark, Peter Gehler, and Bernt Schiele. Occlusion patterns for object class detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3286–3293. IEEE, 2013.

[42] Vincent Rabaud and Serge Belongie. Counting crowded moving objects. In *CVPR*, 2006.

[43] Sitapa Rujikietgumjorn and Robert T Collins. Optimized pedestrian detection for multiple and occluded people. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3690–3697. IEEE, 2013.

[44] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[45] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christopher Bregler. Efficient object localization using convolutional networks. *arXiv preprint arXiv:1411.4280*, 2014.

[46] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2014.

[47] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *TCSVT*, October 2008.

[48] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1879–1886. IEEE, 2011.

[49] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, pages 734–741, 2003.

[50] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[51] Yang Wang, Duan Tran, and Zicheng Liao. Learning hierarchical poselets for human parsing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1705–1712. IEEE, 2011.

[52] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *PAMI*, 19(7):780–785, July 1997.

[53] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007.

[54] Mengyao Zhai, Lei Chen, Jinling Li, Mehran Khodabandeh, and Greg Mori. Object detection in surveillance video from dense trajectories. *Proceedings of 14th IAPR International Conference on Machine Vision Applications 2015*, pages 535–538, 2015.

[55] S. Zhang, C. Bauckhage, and A. Cremers. Informed haar-like features improve pedestrian detection. In *CVPR*, 2014.

[56] Yuting Zhang, Kihyuk Sohn, Ruben Villegas, Gang Pan, and Honglak Lee. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. *arXiv preprint arXiv:1504.03293*, 2015.

[57] Long Zhu, Yuanhao Chen, Alan Yuille, and William Freeman. Latent hierarchical structural learning for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1062–1069. IEEE, 2010.