Mining Semantic Context Information for Intelligent Video Surveillance of Traffic Scenes

Tianzhu Zhang, *Member, IEEE*, Si Liu, *Student Member, IEEE*, Changsheng Xu, *Senior Member, IEEE*, and Hanqing Lu, *Senior Member, IEEE*

***Abstract*—Automated visual surveillance systems are attracting extensive interest due to public security. In this paper, we attempt to mine semantic context information including object-specific context information and scene-specific context information (learned from object-specific context information) to build an intelligent system with robust object detection, tracking, and classification and abnormal event detection. By means of object-specific context information, a cotrained classifier, which takes advantage of the multiview information of objects and reduces the number of labeling training samples, is learned to classify objects into pedestrians or vehicles with high object classification performance. For each kind of object, we learn its corresponding semantic scene-specific context information: motion pattern, width distribution, paths, and entry/exist points. Based on this information, it is efficient to improve object detection and tracking and abnormal event detection. Experimental results demonstrate the effectiveness of our semantic context features for multiple real-world traffic scenes.**

***Index Terms*—Event detection, Gaussian mixture model (GMM) and graph cut, object classification, object detection, object tracking, video surveillance.**

I. Introduction

In recent years, there has been an increasing demand for automated visual surveillance systems [1]–[5]: more and more surveillance cameras are used in public areas such as airports, banks, malls, and subway stations. However, they are not optimally used due to the manual observation of the output, which is expensive and unreliable. Automated surveillance systems aim to integrate real-time and efficient computer vision algorithms in order to assist human operators. This is an ambitious goal which has attracted an increasing amount of researchers to solve commonly encountered surveillance problems of object detection, object classification, object tracking, and abnormality detection [6]–[9] over the years. In this paper, we attempt to solve these problems by mining semantic context information.

Manuscript received August 06, 2011; revised September 08, 2011, October 31, 2011, and January 25, 2012; accepted May 19, 2012. Date of publication September 11, 2012; date of current version December 19, 2012. This work was supported in part by 973 Program 2012CB316304 and 2010CB327905 and the National Natural Science Foundation of China under Grant 61070104 and Grant 6127239. Paper no. TII-11-398.

T. Zhang is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the China-Singapore Institute of Digital Media, Singapore 119613, Singapore.

S. Liu, C. Xu, and H. Lu are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the China-Singapore Institute of Digital Media, Singapore 119613, Singapore (e-mail: sliu@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2012.2218251

Object detection is a basic task in video surveillance [10]. In the situation of stationary cameras, background modeling [6], [11]–[13] is a widely used technique to extract the moving pixels (foreground). If there are few objects in the scene, each connected component of the foreground (blob) usually corresponds to an object; this kind of blob is denoted as single-object. However, it is common that several objects form one big blob, which is called multi-object, because of the angle of the camera, shadow, and moving objects near each other. Since a multi-object is detected as one foreground, it is difficult to obtain the appearance feature of each single object. Therefore, it is difficult to classify and track the objects. A number of works have been proposed to solve the crowd segmentation problem, which emphasized locating individual humans in a crowd. In [14] and [15], head detection is used to help locate the position of humans. Dong *et al.* [16] proposd a novel example-based algorithm that maps a global shape feature by Fourier descriptors to various configurations of humans directly and use locally weighted average to interpolate for the best possible candidate configuration. In addition, they use dynamic programming to mitigate the inherent ambiguity. Zhao and Nevatia [15] use human shapes to interpret foreground in a Bayesian framework. However, these methods are not appropriate for segmenting a group of objects into individual objects. Because directions of motion of objects are different, their postures will change, which may cause these features not to be feasible. In addition, objects in a group may have similar color, texture, and shape features. To solve this problem, we propose a method based on scene-specific context features, which reflect motion rules of objects, including direction of motion and size of object at a certain location.

Object classification is to classify moving objects into semantically meaningful categories [17]. This recognition task is difficult, due to object with diverse visual appearances, which results in large intra-class variations. In recent years, much attention has been attracted on classifying object after object detection. Most previous approaches in this area [18] often use shape and motion information, such as area size, compactness, bounding box, and speed. However, object shapes in video may change drastically under different camera view angles. In addition, the detected shapes may be noised by shadow or other factors. Another important feature is the appearance-based method [19] to achieve robust object classification in diverse camera-viewing angles. However, due to low resolution, shadow, and different viewing angles, classifying objects with only one of these features is not sufficient in video surveillance. Another problem for object classification is how to reduce the number of labeling samples. Currently, the popular method is to adopt a

semi-supervised learning algorithm from a combination of both labeled and unlabeled data. A typical semi-supervised learning algorithm is the cotraining approach proposed by Blum and Mitchell [20]. The basic idea is to train two classifiers on two independent "views" (features) of the same data, using a relatively small number of examples. These classifiers then go through unlabeled examples, label them, and add the most confident predictions to the labeled set of the other classifier. In other words, the classifiers train each other using the unlabeled data. Some work [20], [21] has proved that cotraining can find a very accurate classification. Inspired by the cotraining idea, we propose an unsupervised learning method by combining multiple features with small labeled data for training two classifiers, which are adopted to classify a foreground into a pedestrian or vehicle. The classifiers collaboratively classify the unlabeled data and use this newly labeled data to update each other. In our algorithm, classifiers are not pre-trained, and two relatively independent features are used: object-specific context features and multiblock local binary pattern (MB-LBP) features as the object representation. Each feature is used to train a classifier, and their outputs are combined to give the final classification results. Experiments demonstrate that cotraining can generate an accurate classifier conveniently and effectively.

Object tracking and **abnormal event detection** are another two important tasks in video surveillance [6], [8], [22]–[27]. We learn the scene model by using the observations of tracks over a long period of time. Based on the learned information, we can detect abnormal events, improve object tracking, and help guide vehicles [28]. Recently, many approaches [29]–[32] have been proposed to learn motion patterns. Some of them are based on trajectory analysis [29], [33], [34]. These methods can be categorized into two classes: spatial distance-based methods [34], [35] and spatial distribution-based methods [29]. Spatial distance-based methods take only the pairwise similarities between trajectories. The proposed trajectory similarities or distances include Euclidean distance [35], Hausdorff distance and its variations [34], and dynamic time warping (DTW) [36]. These approaches have several drawbacks: they lack a probabilistic explanation for abnormality detection, require the cluster number in advance, have a high computational cost, and may not well approximate the true similarity. Therefore, spatial distribution-based methods are proposed by Wang *et al.* [29] to avoid these drawbacks. Wang *et al.* [29] use the distributions of observations (positions and moving directions of objects) on the trajectories for trajectory analysis, but do not take into account the integrity of each trajectory. Based on the definition in [29], the continuity of trajectory is ignored. After clustering trajectories, semantic scene models are obtained for each cluster. Paths can be detected by modeling the spatial extents of trajectory clusters [33], [34]. Entry and exit points are detected at the ends of paths based on the velocity distribution [34]. Makris and Ellis [33] detect these points from start/end points of trajectories by the Gaussian mixture models (GMM).

The remainder of this paper is organized as follows. The proposed approach is described in details in Section II. How to learn scene-specific context information is introduced in Section III, and how to use the learned information for several tasks in video surveillance is discussed in Section IV. Experimental results are reported and analyzed in Section V. Finally, we discuss our method and conclude the paper in Section VI.

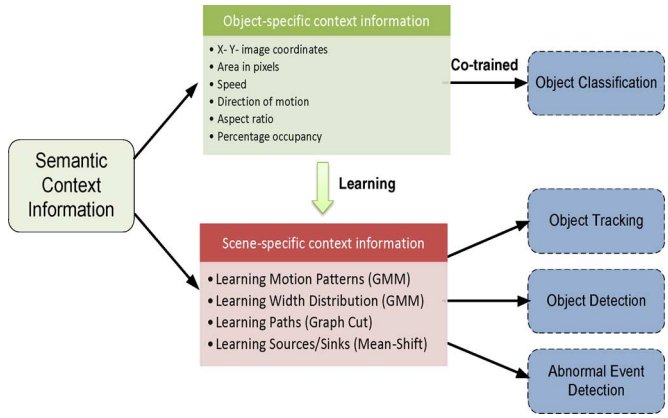


Fig. 1. Proposed framework of mining semantic context information.

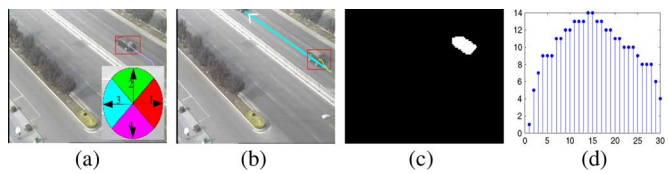


Fig. 2. Information of a detected vehicle. (a) Vehicle is detected with a bounding box. (b) Direction of motion of the vehicle. (c) Binary mask of the vehicle. (d) Vertical projection histogram of the vehicle; the number of bins is the width of the vehicle.

II. Our Approach

The framework of our method is shown in Fig. 1. Semantic context information includes object-specific context information and scene-specific context information. Object-specific context information contains x-, y- image coordinates, area in pixels, speed, direction of motion, aspect ratio, and percentage occupancy. Scene-specific context information is learned with the object-specific context information, and we consider four primary features: motion patterns of objects, width of object, paths, and sources/sinks. Then, the semantic context information is adopted to improve object detection, classification and tracking, and detect abnormal events.

Real-time background subtraction and tracking [37] is used to detect and track the moving objects. For each foreground object, it is easy to obtain its object-specific context information by object detection and tracking, as shown in Fig. 2. By trajectory analysis, GMM is adopted to learn object motion patterns and width distribution, and the graph cut algorithm is used to group similar motion patterns to get paths. Then, trajectories are further clustered. For each cluster of trajectories, entry/exit points and primary trajectories are learned by mean-shift-based multiple data mode-seeking algorithm. Based on the learned information, we improve object classification, detection and tracking, and abnormal event detection. The proposed method is verified on extensive real video data, and the results are encouraging. Different from the existing work [30], [31], [38], which emphasized how to learn motion patterns, this work

focuses on how to mine semantic context information to help several tasks in video surveillance. The contributions of our work are summarized as follows.

1) We propose a novel method to mine semantic context information to improve object detection, classification and tracking, and abnormal event detection.
2) Object classification is improved by fusing different features (object-specific context information and appearance information) and enlarging unlabeled samples under a co-training framework.
3) Scene-specific context information is efficiently and effectively learned by use of GMM and graph cut algorithm.

III. Learning Scene-Specific Context Information

Scene-specific context features reflect the properties of objects in the scene image and can be learned from long-term observations, which can be used to distinguish objects. It is time-consuming and needs a lot of storage space to obtain these features for each pixel in the scene image. Adjacent pixels in the scene image have similar scene context features; therefore, it is viable to cut the scene into $R \times C$ blocks, where R is the number of rows and C is the number of columns. The size of each block is relatively small, hence the motion pattern and size of a moving object in a certain block are considered to be constant. Therefore, the method to learn scene context features based on each block is feasible.

We consider four primary scene-specific context features: motion patterns of objects, width of object, paths, and sources/sinks. Motion patterns of objects can be obtained by analyzing their trajectories, and then the direction of motion in each block can be obtained. Based on the direction in each block, width distribution in each block can be learned by using the width of the vehicle which is extracted by projecting its binary mask onto the direction perpendicular to the direction of motion in each block. Moreover, the paths and sources/sinks are also learned based on motion patterns. The technical details are elaborated below.

A. Learning Motion Patterns for Each Block

A trajectory can be obtained by tracking the centroid of an object and described as $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, where (x_n, y_n) is a point in the scene image described by use of the image coordinate. In general traffic scenes, trajectory of a vehicle is not complicated, thus it is reasonable to use quadratic curve $(y = a \times x^2 + b \times x + c)$ to describe the trajectory. For a tracked object, all points from start point to end point are collected to calculate the parameters (a, b, c) by least squares fit to the y values. Moving direction of object (v) is quantized into four directions as in Fig. 2(a). The parameters (a, b, c, v) are features of a trajectory.

Objects can be classified into vehicles or pedestrians, and there are two types of trajectories. One belongs to vehicles, and the other belongs to pedestrians. For each type of trajectory, the motion patterns of each block can be viewed as Gaussian distributions from statistic point of view. Because each block may contain many motion patterns, we adopt the multiple Gaussian models to represent them. There are four advantages to learn motion patterns by the GMM algorithm. First, multiple Gaussian models are sufficient to describe each block which may contain many various motion patterns. This is because the number of traffic rules is limited, which causes the number of motion patterns in each small block to be limited. Second, outlier trajectories can be removed by updating the weight of Gaussian model, hence primary motion patterns can be learned from long-term observations. Third, the weight of Gaussian model can be viewed as the importance of its corresponding motion pattern, hence the number of important activities will be known. Finally, the computational cost is low. Experiments demonstrate that the GMM algorithm is efficient.

Our algorithm can be described as follows. Each block in the scene is modeled by a mixture of K Gaussian distributions for trajectory parameters. For a certain block, the series of trajectories $\{T_t = (a_t, b_t, c_t, v_t)\}_{t=1}^{N}$ are obtained. Here, (a_t, b_t, c_t, v_t) are parameters of a trajectory T_t. They are used to learn the parameter distribution of blocks which the objects have passed. The probability that a certain block has a value of T_t at time t can be written as

$$P(T_t) = \sum_{i=1}^{K} w_{i,t} \times \eta(T_t, u_{i,t}, \Sigma_{i,t}) \tag{1}$$

where $w_{i,t}$ is the weight parameter of the ith Gaussian component at time t, $\eta(T_t, u_{i,t}, \Sigma_{i,t})$ is the ith normal distribution of component with mean $u_{i,t}$ and covariance $\Sigma_{i,t}$. Here, $\Sigma_{i,t}$ is assumed to be diagonal matrix. Although this may not be true in all of the cases, the assumption allows us to avoid a costly matrix inversion at the expense of some accuracy:

$$u_{i,t} = \left(u_{i,t}^a, u_{i,t}^b, u_{i,t}^c\right)^T \tag{2}$$

$$\Sigma_{i,t}^{\frac{1}{2}} = \begin{pmatrix} \sigma_{i,t}^a & 0 & 0 \\ 0 & \sigma_{i,t}^b & 0 \\ 0 & 0 & \sigma_{i,t}^c \end{pmatrix}. \tag{3}$$

The K distributions are ordered based on the fitness value $w_{i,t}$. Parameters u and σ for unmatched distributions remain the same. The first Gaussian component that matches the test trajectory will be updated by the following update equations:

$$w_{i,t} = (1 - \alpha) w_{i,t-1} + \alpha(M_{i,t}) \tag{4}$$

$$u_{i,t} = (1 - \rho) u_{i,t-1} + \rho T_t \tag{5}$$

$$\sigma_{i,t}^2 = (1 - \rho)\sigma_{i,(t-1)}^2 + \rho(T_t - u_{i,t})^T (T_t - u_{i,t}) \tag{6}$$

$$\rho = \alpha \eta(T_t | u_{i,t}, \sigma_{i,t}) \tag{7}$$

where $\sigma_{i,t} = (\sigma_{i,t}^a, \sigma_{i,t}^b, \sigma_{i,t}^c)^T$, $M_{i,t}$ is 1 for the model which matched and 0 for the remaining models, and $1/\alpha$ defines the time constant which determines change. If none of the K distributions matches the trajectory, the component with the minimum weight is replaced by a distribution with the current value (a_t, b_t, c_t) as its mean, the v_t as its moving direction, an initially high variance, and a low weight parameter. In our experiments, the number of primary motion pattern of a small block is no more than 3 in the traffic scene, therefore, K is manually set to be 3 for simplicity. For our method, this parameter is not critical to decide the number of motion patterns in each block. This is because we have the weight $(w_{i,t})$ for each Gaussian component (motion pattern), which can help us decide the importance of each motion pattern and the number of motion patterns in

each block as introduced in Section III-C. α is manually set to be 0.1, the initial high variances of (a, b, c) are (0.05, 0.2, 20), and the low weight is 0.05 by experience.

B. Learning Width Distribution for Each Block

Based on the learned motion patterns, motion direction for each block can be obtained (the parameters a and b), and then width distribution for each block can be learned. For a foreground with width w_t in block (x_0, y_0) at time t, the width of the foreground is used to learn the width distribution for the block.

In traffic scene, a foreground may be a single-vehicle blob or a multivehicle blob, their width may have significant difference. Therefore, in each block, the probabilistic distribution of the width is modeled as a GMM. We take each of the Gaussian components as one of the underlying width distribution and update the GMM parameters with adaptive weights in an online way just as the process of learning motion patterns for each block. The parameters (mean w_u and variance w_σ) of Gaussian component with the maximum weight are considered as the features of each block.

C. Learning Paths for Scene

Paths are composed of blocks with similar motion patterns. Therefore, we can obtain paths by grouping the patterns with the commonly used clustering algorithm. However, the algorithms, for example, the K-means algorithm, do not consider the spatial relations between local blocks. Actually, for two neighboring blocks, it is possible that they contain similar motion patterns. Thus, it is necessary to consider such spatial relations when grouping the motion patterns together, which yields the following graph-based algorithm.

Naturally, we can take each local block as a node, and two neighboring blocks in horizontal and vertical directions can be connected together. In terms of Markov random field (MRF), we consider minimizing the following energy function:

$$E(L) = \sum_{p \in S} D_p(L_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(L_p, L_q) \tag{8}$$

where S is the block lattice, $\mathcal{N}$ is the pairwise neighborhood system, the label set is {0,1}, $D_p(L_p)$ denotes the cost of the block p to be labeled as L_p, and $V_{p,q}(L_p, L_q)$ encourages spatially neighboring blocks to have similar labels.

Now, a core task is to calculate the terms of $D_p(L_p)$ and $V_{p,q}(L_p, L_q)$ in (8). Note that we have no prior information about the patterns. To assign values to $D_p(L_p)$, we first extract the primary motion patterns in the scene.

Since all motion patterns ($\mathbb{G}$) learned by GMM algorithm are collected into $\mathbb{G} = \{\vec{g}_{ij}^k | i = 1, 2, \cdots, R, j = 1, 2, \cdots, C, k = 1 \cdots K.\}$, where $\vec{g}_{ij}^k = (a_{ij}^k, b_{ij}^k, c_{ij}^k, v_{ij}^k)^T$ is the kth motion pattern of the block (i, j), and then we can consider to take the components from $\mathbb{G}$ as the primary motion patterns. These patterns will be used as references to calculate the energy term.

As for a scene, there are only a few primary motion patterns. The weight of a Gaussian model reflects the importance of a motion pattern. Therefore, a Gaussian model $\vec{g}_{ij}^k$ is selected as a primary motion pattern if its weight $w_{ij}^k > Th$. Th is a threshold which is manually set to be 0.85 in our experiments throughout our evaluation. In this way, all of the primary motion patterns ($\mathbb{G}_m$) are extracted. For a motion pattern in $\mathbb{G}_m$, which is considered as a reference, all of the primary motion patterns ($\mathbb{G}_m$) are viewed as two clusters, denoting them as $\vec{g}_r$ and $\vec{g}_a$, respectively. $\vec{g}_a$ is the average of motion patterns which belong to $\{\vec{g}_l \| \|\vec{g}_r - \vec{g}_l\| > \lambda \times d_{\text{mean}}, v_r \cdot v_l = 1\}$, where λ is a tuned coefficient and is manually set to be 0.85, d_{mean} is the average distance between the reference and the motion patterns having similar velocity with the reference in $\mathbb{G}_m$. Distance between motion pattern $\vec{g}_1 = (a_1, b_1, c_1, v_1)$ and $\vec{g}_2 = (a_2, b_2, c_2, v_2)$ is defined as follows: if $v_1 \cdot v_2 = 1$, $\|\vec{g}_1 - \vec{g}_2\|^2 = (a_1 - a_2)^2 + (b_1 - b_2)^2 + (c_1 - c_2)^2$, otherwise, $\|\vec{g}_1 - \vec{g}_2\|^2 = Max$, where Max is a large constant number and is set to 99 in our experiments. In fact, we can set it as many large constant numbers, if they are much larger than the normal distance between two motion patterns. Now we can calculate $D_p(L_p)$ as follows:

$$D_p(L_p) = \begin{cases} \frac{d_1}{d_1 + d_2}, & \text{if } L_p = 1 \\ \frac{d_2}{d_1 + d_2}, & \text{if } L_p = 0 \end{cases} \tag{9}$$

where $d_1 = \min_k \|\vec{g}_r - \vec{g}_{ij}^k\|, d_2 = \min_k \|\vec{g}_a - \vec{g}_{ij}^k\|$. d_1 and d_2 represent the similarities between the motion patterns of block (i, j) and $\vec{g}_r, \vec{g}_a$ respectively. Furthermore, $V_{p,q}(L_p, L_q)$ is evaluated as

$$V_{p,q}(L_p, L_q) = \begin{cases} 0, & \text{if } L_p = L_q \\ d_0, & \text{otherwise} \end{cases} \tag{10}$$

where d_0 is a constant to punish a label jumping and is calculated as the standard deviation of $\{d_{ij} | i = 1, 2, \cdots, R; j = 1, 2, \cdots, C\}$, and d_{ij} is $\min_k(\|\vec{g}_r - \vec{g}_{ij}^k\|)$, which represents the minimum distance between motion patterns of block (i, j) and the reference motion pattern.

Finally, for each motion pattern in $\mathbb{G}_m$, it is viewed as a reference, and the graph-cut algorithm [39], [40] is applied to optimize the objective function (8) to obtain its corresponding semantic region. In this way, all of the semantic regions are obtained. Trajectories which fit the same semantic regions are viewed as a cluster. For each cluster, the measure in [34] is taken to extend the distribution of the trajectories to obtain its corresponding paths and mean-shift algorithm [41] is used to obtain the primary trajectory.

D. Learning Sources/Sinks

Two interesting scene structures are locations where vehicles enter or exit the scene. They are called sources and sinks. Trajectories are often broken because of inevitable tracking failures. There are false entry/exit points biasing the estimation of sources and sinks as shown in Fig. 8. In each trajectory cluster, sources and sinks should be on the two ends of the path regions, and mean-shift algorithm is adopted to search these points.

IV. Applications of Semantic Context Information

A. Improvement of Object Classification

To train the classifiers, labeling a large training set by hand can be time-consuming and tedious. The difficulty is the high cost of acquiring a large set of labeled examples to train the two

classifiers. Usually, a collection of a large number of unlabeled examples in most applications has a much lower cost, as it requires no human intervention. Therefore, we adopt a semi-supervised method to learn two classifiers, inspired by the idea of cotraining learning. Two sets of features are predefined and they are relatively independent of each other: 1) object-specific context features, such as position, area in pixels, and velocity and 2) appearance features based on MB-LBP. Two labeled sets are then prepared based on them, each for training one of the classifiers. Each classifier predicts on the unlabeled samples to enlarge the training set of the other.

1) LDA-Based Classifier: Since object-specific context features reflect the properties of objects in the scene image, they can be used to distinguish objects. For a certain scene, these features are robust and useful to classify object [34], [42]. In our work, object-specific context features: x- and y-image coordinates, area in pixels, speed, direction of motion, aspect ratio, and percentage occupancy, are used to find an optimal direction of projection to separate the positive and negative sample with Fisher linear discriminant analysis (LDA). The projection function is defined as $g = w^T r$, where $w = (S^1 + S^2)^{-1}(u^1 - u^2), u^1, u^2$ are the means of the two classes (people and vehicle), and S^1, S^2 are the covariance matrices. The formulation of u^t and $S^t, t = 1, 2$, are as follows: $u^t = (1/n_t)\sum_j^{n_t} r_j, S^t = (1/n_t - 1)\sum_j^{n_t}(r_j - u^r)(r_j - u^r)^T$, where r_j is the jth sample.

2) AdaBoost Classifier: Since the LDA-based classifier constructed by object-specific context features is relevant to the scene, an appearance classifier based on MB-LBP [19] features is adopted to improve the performance of classification. MB-LBP is extended from the original LBP feature [43], which has been proven to be a powerful appearance descriptor with computational simplicity. In addition, this feature is also successfully applied in many low-resolution image analysis tasks. However, it is limited to calculate the information in a small region and has no ability to capture large-scale structures of objects. MB-LBP is developed on image patches divided into subblocks (rectangles) with different sizes. This treatment provides a mechanism to capture appearance structures with various scales and aspect ratios. Intrinsically, MB-LBP is to measure the intensity differences between subblocks in image patches. Calculation on blocks is robust to noises, and light change. At the same time, MB-LBP can be computed very efficiently by using integral images [44]. The feature set of MB-LBP feature is large and contains much redundant information. The gentle AdaBoost [45] is used to select significant features and construct a binary classifier.

3) Cotraining Strategy: In the cotraining framework, the classifiers are trained as follows. For a certain scene, some samples are labeled to train the two classifiers, then the classifiers are used to classify unlabeled examples to obtain their labels and add those newly labeled examples which are confident enough to update the training set for each other. This learning process can be repeated many times. As the LDA-based classifier is relevant to the scene, for a different scene, the AdaBoost classifier is used to label samples to train the LDA-based classifier, then the two classifiers are trained for each other. To ensure the correct classification, from one training data set to another, their appearances change slowly.

The main advantages of this scheme include the following. First, it is a collaborative approach that uses the strength of different views of the object to help improve each other, hence a more robust classification can be achieved. Second, mass manually labeling is avoided. Experiments demonstrate that cotraining can generate accurate classifiers. After training classifiers, we make final classification decision according to the output of the classifier with more confidence.

B. Improvement of Object Detection

For traffic scenes, the vehicle is one primary object, and we take two steps by use of the learned scene-specific context information to improve its detection as in our previous work [42]. First, a classifier is adopted to classify the foreground into single-vehicle or multivehicle objects. The second step is to segment the multivehicle blob into single-vehicle blobs.

1) Bayes Classifier: The width of a foreground at time t is denoted by x_t. The naive Bayes classifier is to decide if the foreground belongs to multivehicle (MV) or single-vehicle (SV). Bayesian decision L is made by

$$L = \frac{p(MV|x_t)}{p(SV|x_t)} = \frac{p(x_t|MV)p(MV)}{p(x_t|SV)p(SV)}. \tag{11}$$

In a general case, we do not know anything about the foreground objects that can be seen nor when and how often they will be present. Therefore, we set $p(MV) = p(SV)$. We decide then that the foreground belongs to an MV blob if

$$p(x_t|MV) > L \times p(x_t|SV). \tag{12}$$

We will refer to $p(x|MV)$ and $p(x|SV)$ as the models. The models are estimated from different training sets denoted as $\mathcal{X}$ and $\mathcal{Y}$, respectively. The estimated models are denoted by $\hat{p}(x|\mathcal{X}, MV)$ and $\hat{p}(x|\mathcal{Y}, SV)$ which depend on the training set as denoted explicitly. We assume that the samples are independent and the main problem is how to efficiently estimate the density function and adapt it to possible changes.

In order to guarantee the performance of Bayes classifier, We use GMM with M components

$$\hat{p}(x|\mathcal{X}, MV) = \sum_{m=1}^{M} \hat{w}_m \hat{\eta}\left(x; \hat{u}_m, \hat{\delta}_m^2 I\right) \tag{13}$$

$$\hat{p}(x|\mathcal{Y}, SV) = \sum_{n=1}^{M} \hat{w}_n \hat{\eta}\left(x; \hat{u}_n, \hat{\delta}_n^2 I\right) \tag{14}$$

where $\hat{u}_1, \cdots, \hat{u}_M$ are the estimates of the means and $\hat{\delta}_1, \cdots, \hat{\delta}_M$ are the estimates of the variances that describe the Gaussian components. In our experiment, M is set to be 2 considering that there may be two different kinds of foregrounds (MV and SV). The covariance matrices are assumed to be diagonal, and the identity matrix I has proper dimensions. The parameters are updated as the same as (4)–(6).

For a dataset, the Bayes classifier is initialized online using scene-specific context features. For a foreground with width x_t at block (x_0, y_0) in a scene image, scene-specific context fea-

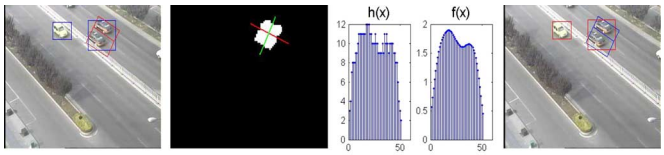


Fig. 3. (a) Results of GMM detector denoted by blue bounding boxes. (b) Binary mask of the blob. The red and green lines indicate two different directions of motion, respectively. (c) Vertical projection histograms of the blob. (d) MV blob is segmented by blue boxes with our algorithm.

tures (the mean w_u and variance w_σ of width) of this block can be used to classify the foreground into MV blob or SV blob. In practice, a foreground is usually an SV blob. Therefore, the primary distribution of width in each block reflects width distribution of SV blob. If $x_t - w_u/w_\sigma > Th$ holds true, the foreground is an MV blob, and the parameters of the model $\hat{p}(x|\mathcal{X}, MV)$ are updated; otherwise, the model $\hat{p}(x|\mathcal{Y}, SV)$ is updated. Th is a given threshold and set to be 2.0 experimentally. Once these models are learned, they can be used to label the moving objects.

2) MV Blob Segmentation: Shape feature has been used to segment and localize individual humans in a crowd [14]–[16], [46]. However, it is difficult to simply use shape feature to segment vehicles in video surveillance. For example, Fig. 3(b) explains the reason. If we do not know the motion direction of a vehicle, we are not sure whether a square-shaped blob contains multiple vehicles or not. As for a vehicle, its length is longer than its width. If the moving direction is the direction of the green line, the foreground looks like an SV. If the moving direction is the direction of the red line, the foreground looks like an MV. In addition, vehicles in a blob may have similar color, texture, and shape features, therefore, it is difficult to segment a blob into individual vehicles using these features. In a fixed scene, scene-specific context features (such as direction of motion and width distribution of vehicles) are stable. These features are helpful to segment MVs. Therefore, we propose a novel method based on scene-specific context features to improve vehicle detection accuracy.

For an MV blob, its vertical projection histogram $h(x)$ can be obtained by projecting its binary mask onto the direction that is perpendicular to the direction of motion. To obtain the junctions of vehicles conveniently, the vertical projection histogram is smoothed to construct $f(x) = (1/N)\sum_{i=1}^{N}(h(x_i) \times \exp(-(x - x_i)^2/w_u))$, where w_u is the mean width of vehicle in the block which the foreground is inside. Based on the $f(x)$, we can segment the MV blobs as shown in Fig. 3(d).

C. Improvement of Object Tracking

Motion of an object can be predicted using the learned motion patterns. Let $T_0 = \{(x_1, y_1), (x_2, y_2), \cdots, (x_k, y_k)\}$ be the initial part of a motion trajectory with k points. The probability $P(T_0|g_m)$ of T_0 under each motion pattern g_m is calculated as formulated in (15). $P(T_0|g_m)$ is just the probability that the object is expected to move along the trajectory represented by motion pattern g_m. The trajectory represented by the motion pattern with the highest probability is chosen as the most probable one along which the object is expected to move. If $P(T_0|g_m)$ is very small, g_m is rejected as a possible trajectory for the object:

$$P(T_0|g_m) = \exp\left(-\sum_i \left(a_m \times x_i^2 + b_m \times x_i + c_m - y_i\right)^2\right) \tag{15}$$

where $g_m = (a_m, b_m, c_m)^T$ and $g_m \in \mathbb{G}_m$. The motion pattern g_m that has the maximum likelihood with trajectory T_0 can be used to help improve object tracking. For example, we can use the k points to estimate the speed v of an object. Given the object in point (x_{t_1}, y_{t_1}) at time t_1, and $x_{t+2} = x_{t_1} + v \times (t_2 - t_1)$, based on the motion pattern $g_m, y_{t+2} = a_m \times x_{t+2}^2 + b_m \times x_{t+2} + c_m, (x_{t+2}, y_{t+2})$ can be predicated. Moreover, by use of moving object detection, we can also refine the object tracking.

According to the Bayes rule, the probability of g_m given T_0 is calculated by

$$P(g_m|T_0) = \frac{P(T_0|g_m)P(g_m)}{\sum_m P(T_0|g_m)P(g_m)}, \quad m = 1, 2, \ldots |G_m|$$

where $P(g_m)$ is assumed to be a constant number with different g_m. We can find the pattern g_m to which T_0 corresponds by $m^* = \arg\max_m P(g_m|T_0)$.

D. Abnormal Event Detection

Based on the learned motion patterns, our method can detect abnormal events that are defined as vehicles breaking the traffic rules by use of their trajectories. Given a trajectory $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, we calculate the probability of T under each motion pattern and look for the motion pattern g_m that has the maximum likelihood with trajectory T : $m^* = \arg\max_m P(T|g_m)$. If the probability $P(T|g_m)$ of T under the motion pattern g_{m^*} is less than a threshold th_{m^*}, the trajectory is treated as abnormal. We use the probability of each of the trajectories which correspond to g_{m^*} given motion pattern g_{m^*} to calculate the threshold th_{m^*}. For each of these trajectories T, we calculate the probability $P(T|g_{m^*})$ of T under motion pattern g_{m^*}. We take the minimum of all $P(T|g_{m^*})$ as the threshold th_{m^*} : $th_{m^*} = \min_T P(T|g_m^*)$. In this way, each motion pattern has a threshold. Therefore, we can acquire a threshold set: $\{th_1, th_2, \ldots th_m\}$, where each threshold th_m is for each motion pattern g_m.

V. Experimental Results

Here, we first describe the datasets used for our proposed method. We then present the experimental results for each task.

A. Datasets

There are few publicly available datasets suitable for our tasks. Therefore, we attempt to adopt many videos from real-world traffic scenes to evaluate our methods. To demonstrate the effectiveness of motion pattern learning, we run our algorithm on many videos from different traffic scenes. In this work, the results are shown in two typical scenes, which include the primary traffic scenes: straight road and crossroad. For object classification, we collect 12 videos (about 4 h) from six different traffic scenes to evaluate the performance of our

Fig. 4. Exemplar frames from our datasets. Each frame shows one real-world traffic scene.

proposed semi-supervised learning classifier. For object detection evaluation, we collect 18 videos (about 6 h) from eight traffic scenes. We also adopt videos from crossroad to show the results of object tracking and videos from two scenes to detect abnormal events. All of these videos are from real-world traffic scenes and include illumination changes, occlusions, a variety of object types, shadows, and different environmental effects. Some exemplar images from these videos are shown in Fig. 4. In addition, we also show results of learning paths and improvement of object detection on i-LIDS parked vehicle detection dataset [47].

To evaluate the effectiveness of our proposed method, we compare with some existing methods. For motion pattern learning, we compare with [34] and [48]. For object classification, we compare with LDA-Based classifier and AdaBoost Classifier [19]. For object detection, we compare with GMM [6]. All of these results on the datasets are efficient to demonstrate the effectiveness of our proposed method.

B. Learning Scene-Specific Context Features

Here, we will introduce the results of scene-specific context information including motion patterns, width distribution, paths, and sources/sinks.

1) Results of Learning Motion Patterns and Width Distribution: Motion pattern and width of a vehicle are adopted to learn scene-specific context features for each block. As shown in Fig. 2, a vehicle is tracked from the entry point to the exit point, then the trajectory is fit to get motion pattern for each block which the vehicle has passed. As for the blocks the vehicle has passed, the direction of motion (motion pattern) in each block is used to get width of the vehicle, then the width is used to learn the width distribution for the block. Some results are illustrated in Fig. 5. The scene image is cut into multiblocks in Fig. 5(a) to learn those context features. In our experiments, R and C are both set to be 8 for a 320 × 240 image resolution. The direction of motion and mean width of a vehicle in each block is displayed in Fig. 5(b) and (c), respectively. For blocks which the vehicle has not passed, their features are 0. The GMM algorithm updates weight in an online way, which guarantees that the primary distribution for each block can be learned. This results exhibit that our approach is effective.

2) Results of Learning Paths: We cluster blocks with similar motion patterns and group the trajectories. Before clustering, outlier trajectories must be removed. Usually, these are noisy

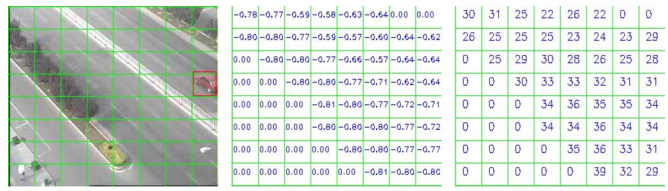


Fig. 5. (a) Scene is cut into 8 × 8 blocks. (b) Value of direction of motion b in each block. Because the road is straight line, the parameter a is 0. (c) Mean of width in each block.

TABLE I
FIRST ROW (TN) IS THE NUMBER OF TRAJECTORY. THE SECOND AND THIRD ROWS ARE THE CORRESPONDING COMPUTATION TIME OF I AND II, THE UNIT IS SECOND. THE FOURTH ROW IS THE CORRESPONDING TIME OF LEARNING MOTION PATTERNS WITH THE GMM ALGORITHM

TN	200	400	600	800	1000
I (seconds)	1112	4138	11687	17578	28072
II (seconds)	2	5	12	23	38
III (seconds)	**4.8**	**5.6**	**6.0**	**7.2**	**8.3**

trajectories caused by tracking or classification errors, anomalous trajectories, e.g., a car drives out of the way. In visual surveillance, these may be of particular interest, and, as expected, our algorithm can detect them. For each scene, the GMM algorithm is adopted to learn motion patterns for a long time. If the trajectory's parameters are similar to the motion patterns of blocks which the object has passed and the motion patterns have a high weight, the trajectory is received. Otherwise, the trajectory is viewed as a noisy trajectory and deleted. In this way, 364, 417, and 173 trajectories are selected in scenes S1 and S2 and on i-LIDS parked vehicle detection dataset, respectively.

Three trajectory clustering methods are compared in our experiments. For clarity, they are described as I, II, and III.

I) As mentioned in [34], the modified Hausdorff distance is viewed as trajectory similarity and uses spectral clustering [48].
II) Based on the parameters of trajectories, Euclidean distance is considered as trajectory similarity and uses spectral clustering [48].
III) Cluster trajectories based on their distributions. Method I is more time-consuming than methods II and III.

Table I gives a quantitative comparison in computational cost on a P4 3.0-GHz CPU. Our method III takes two steps to obtain semantic regions. The first step is cutting a scene image into multiple blocks and learning motion patterns for each block by the GMM algorithm. The corresponding computation time is showed in Table I. The second step is clustering these motion patterns. For scene S1 and S2, the scene image 320 × 240 is cut into 16 × 16 blocks. For i-LIDS parked vehicle detection dataset, the scene image 576 × 720 is cut into 16 × 30 blocks. Blocks having similar motion patterns are clustered to construct semantic region. Based on the weight of Gaussian model, 9, 18, and 2 motion patterns are selected as the primary motion patterns for scenes S1 and S2 and i-LIDS parked vehicle detection dataset, respectively. The corresponding computation time with the graph-cut algorithm is 10, 19, and 20 s, respectively. Because some primary motion patterns are similar, it causes that their corresponding semantic regions may have the same blocks.

Fig. 6. Results of clustering motion patterns. (a) Six semantic regions in scene S1. (b) Eight semantic regions in scene S2. (c) Two semantic regions on i-LIDS parked vehicle detection dataset.

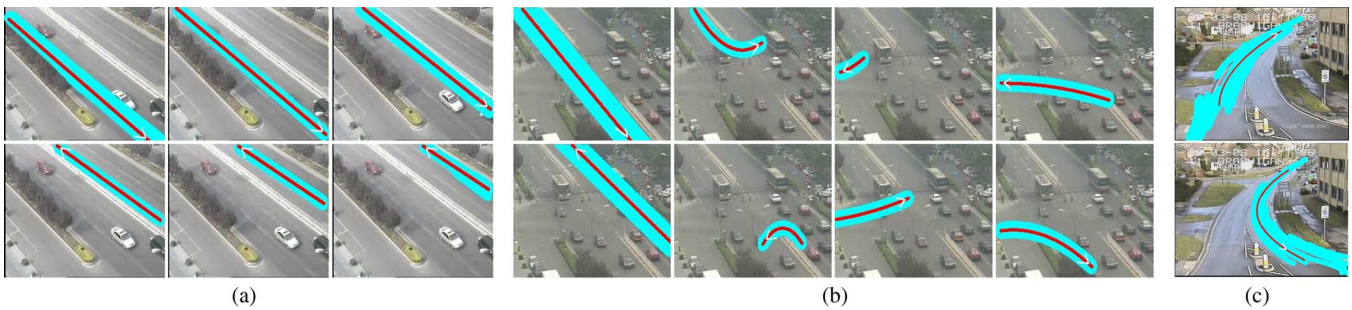


Fig. 7. (a)–(c) Results of paths in scenes S1 and S2 and on i-LIDS parked vehicle detection dataset, respectively. The white arrows are the moving directions of objects, and the red curves are the primary trajectories.

After removing the same semantic regions, all of the primary semantic regions are obtained. Some results are shown in Fig. 6. In scenes S1 and S2, and on i-LIDS parked vehicle detection dataset, there are six, eight, and two semantic regions of vehicles, respectively. Each of them represents a primary motion pattern.

Based on the semantic regions and their corresponding motion patterns, trajectories which fit the same semantic regions are considered as a cluster. For the 364 trajectories in scene S1, there are six clusters. The ground truth annotation of each cluster is labeled manually. Recall and Precision are used to measure the performance. The mean recall and precision of scene S1 are 94.48%, 91.63% by method I, 90.28%, 85.37% by method II, and 98.58%, 96.45% by method III, respectively. These results show that our method III based on spatial distribution of trajectories performs the best. For each cluster of trajectories, the corresponding path region is obtained by thresholding the density distribution. The learned paths of scenes S1 and S2 and the i-LIDS parked vehicle detection dataset are shown in Fig. 7. The red lines are the primary trajectories, which represent the primary motion patterns. The white arrows are the moving directions of objects.

To make a quantitative comparison, the statistical results of i-LIDS parked vehicle detection dataset are illustrated in Table II. For the 173 trajectories, there are two clusters. The ground truth annotation of each cluster is labeled manually. There are 97 and 76 trajectories for clusters 1 and 2, respectively. Recall and Precision are used to measure the performance. These results show our method III based on spatial distribution of trajectories performs the best.

3) Results of Learning Sources/Sinks: The results of sources/sinks are shown in Fig. 8. From these figures, we can see that there are many entry/exit points. In each trajectory cluster, sources and sinks can be searched by the mean-shift algorithm.

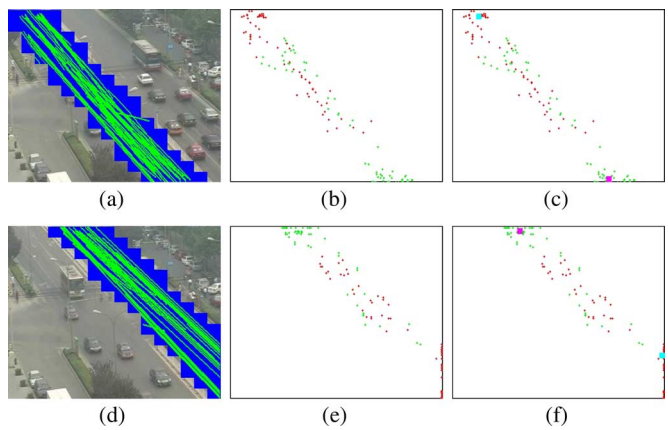


Fig. 8. Two examples about the learned sources/sinks. (a), (d) Two clusters of trajectory. (b), (e) Entry/exit points. (c), (f) Learned sources/sinks denoted as purple and cyan.

TABLE II
PRECISION AND RECALL OF THE THREE METHODS ON I-LIDS PARKED VEHICLE DETECTION DATASET. TP IS TRUE POSITIVE, FN IS FALSE NEGATIVE, AND FP IS FALSE POSITIVE

Method	Cluster	TP	FN	FP	Recall	Precision
I	1	87	10	13	89.7%	87.0%
	2	63	13	10	82.9%	86.3%
II	1	77	20	23	79.4%	77.0%
	2	53	23	20	69.7%	72.6%
III (Our)	1	93	4	3	95.9%	**96.9%**
	2	173	3	4	96.1%	**97.3%**

TABLE III
CLASSIFICATION RESULTS OF THE FOUR CLASSIFIERS. OUR CLASSIFIER IS BEST BECAUSE IT FUSES MULTIPLE FEATURES AND ENLARGES TRAINING SET FROM UNLABELED SAMPLES

Scene	S1	S2	S3	S4	S5	S6
LLC calssifier [49]	89.4%	90.6%	90.2%	93.5%	87.6%	88.7%
LDA-Based classifier	87.1%	88.3%	91.3%	91.5%	82.5%	84.1%
AdaBoost calssifier [19]	91.1%	87.3%	89.8%	90.3%	80.5%	85.3%
Our	**98.2%**	**97.3%**	**96.6%**	**97.4%**	**96.8%**	**97.8%**

Fig. 9. Some segmentation results in the eight scenes and on i-LIDS parked vehicle detection dataset. The red boxes are the results of GMM detector. The blue boxes segment the MV blobs into SV blobs using our algorithm.

C. Results of Object Classification

We compare three classifiers with our proposed cotraining classifier in six scenes. The three classifiers are denoted as the AdaBoost classifier [19], the LDA-based classifier, and the LLC classifier [49]. The AdaBoost classifier [19] is trained with 20 213 positive samples (pedestrians) and 41 934 negative samples (vehicles) labeled manually. These samples are obtained by normalizing blobs to 20×20 pixels and collected per 10 frames to reduce the correlation. The LDA-based classifier is trained with 12 000 positive samples and 35 000 negative samples for each scene using scene context features. The LLC classifier is a locality-constrained linear coding (LLC) [49] method for image classification, which has shown a better performance than existing approaches. LLC utilizes the locality constraints to project each descriptor into its local-coordinate system, and the projected coordinates are integrated by maximum pooling to generate the final representation. Then, a linear SVM classifier is trained based on the learned representation. In this work, we use the histogram of oriented gradient (HOG) [50] descriptor. In our setup, the HOG features are extracted from patches densely located by every pixel on the training samples, under two scales, 4×4 and 8×8, respectively. Based on these feature points, we cluster a codebook with K-means, and the number of cluster is set to be 1024. We adopt 4×4, 2×2 and 1×1 subregions for spatial pyramid matching (SPM) [51], and we use the maximum pooling to generate the final representation. During LLC processing, the number of neighbors was set to 5 as the paper [49]. For each scene, we adopt the same training samples as the LDA-based classifier. Our classifiers are initialized with 2720 positive samples (pedestrians) and 6716 negative samples (vehicles) labeled manually in a data set, then they are trained with our cotraining framework. For a different training set, the AdaBoost classifier is used to classify unlabeled examples to obtain their labels and add those newly labeled examples which are confident enough to update the training set for LDA-based classifier, then the two classifiers are trained for each other. In this way, the training set of our classifiers becomes large and their classification correction rate is rising. In applications, the output of the classifier with more confidence is used to give the final classification decision. The objects in the test set are all not included in the training set. Appearances of objects vary significantly due to shadow, different viewing angles, object merging, and low resolution in video surveillance, thus it is difficult for the AdaBoost classifier and LLC calssifier to have a good performance. Positions of objects affect mostly the classification of the LDA-based classifier, which is specific for each traffic scene. Compared with the AdaBoost classifier [19], the LDA-based classifier and LLC calssifier [49], our proposed cotraining classifier does not require a very large set of labeled training data and achieves more considerable performance in diverse scenes. This is because our method can fuse the strength of different features and increase the training samples by the semi-supervised cotraining method automatically. Table III shows the classification results in detail.

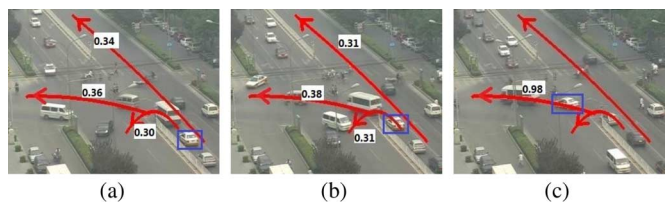
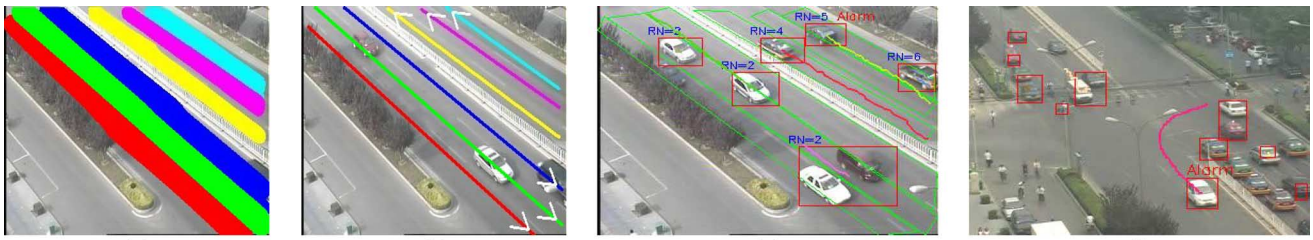


Fig. 10. Motion prediction in the real traffic scene to help object tracking.

D. Results of Object Detection

Moving objects can be reasonably separated with background subtraction and blobs can be obtained. For each blob, we classify it using the Bayes classifier and segment the MV blobs into multiple SV blobs. To test the performance of the classifiers,



(a) (b) (c) (d)

Fig. 11. (a) Six semantic scene models of vehicles in scene S1. (b) Primary trajectories of the six semantic scene models. (c) Lane-merging detection in scene S1. (d) Anomalous trajectory detection in scene S2.

TABLE IV
CLASSIFICATION RESULTS ON TEST DATASET

	Tracks	Correct Classification	Correct Rate
M-Vehicle	1382	1279	**92.6%**
S-Vehicle	756	709	**93.8%**

we collect 2138 vehicle tracked sequences from ten different scenes, and some exemplar frames are shown in Fig. 9. The vehicles in these test sequences are all not included in the training set. A simple voting method to the tracked sequence is used to get a final class label. Table IV shows the classification results. These results indicate that our approach achieves considerable performance in diverse scenes.

For a foreground, once it is classified as an MV blob, the segmentation module will be started up. Supposing that the foreground is inside a certain block, and the direction of the motion and width distribution of vehicles in the block are used to obtain the vertical projection histogram of the foreground, segmented boxes (blue boxes) can be obtained by finding troughs of the vertical projection histogram together with making use of the direction of motion in corresponding block.

Some results of vertical projection histogram $h(x)$ and $f(x)$ are given in Fig. 3(c). We test our algorithm in eight scenes and collect video images randomly. There are 11 365 MV blobs in these images, and 10 797 of them have been segmented into SV blobs correctly. The segmentation correct rate is about 95.0%. On i-LIDS parked vehicle detection dataset, we collect 576 MV blobs, and only 418 blobs are correctly segmented into SV blobs. The correct rate is about 72.6%. From these results on this dataset, we can see that our method cannot obtain good improvement as in the eight scenes. This is because our scene-specific context information-based method is suitable for far-field and medium-field surveillance videos and will obtain better results in these traffic scenes. Some results of segmentation are shown in Fig. 9. In Fig. 9, red boxes are detected by the GMM algorithm, and blue boxes are the results of our segmentation. They validate the performance of segmentation.

E. Results of Object Tracking

Fig. 10 shows an example of prediction in the real traffic scene. The percentage beside a trajectory represents the probability with which the car is expected to move along the trajectory. The car entered the scene from the right-bottom corner and then turned left. Fig. 10(a) shows the three most probable trajectories which the car might follow; in Fig. 10(b), the probability values with which the car would move along these three trajectories are changed; in Fig. 10(c), it tended to turn left with high probability and this motion became clear. To obtain quantitative results, we track objects to get their trajectories and randomly select 213 trajectories for evaluation. For these trajectories, we use our model to predict their future motion in next frames. 197 trajectories are predicted correctly, and the correct rate is about 92.5%. From Fig. 10, we can confirm that our proposed algorithm has a good accuracy in predicting object behaviors.

TABLE V
PRECISION AND RECALL OF THE TWO METHODS ABOUT ABNORMAL EVENT DETECTION. TP IS TRUE POSITIVE, FN IS FALSE NEGATIVE, AND FP IS FALSE POSITIVE

Method	Class	TP	FN	FP	Recall	Precision
Ma	1	13	8	11	61.9%	54.2%
	2	68	11	8	86.1%	89.5%
Mb	1	19	2	4	**85.7%**	**81.8%**
	2	75	4	2	**94.9%**	**96.2%**

F. Results of Abnormal Detection

The learned semantic scene-specific context information can be directly used to real-time detection of abnormal behaviors in traffic scenarios. For scene S1, boundaries of the six semantic scene models of vehicles are described with a rectangle as shown in Fig. 11. The six paths are labeled from 1 to 6, for example, "RN $= 2$" represents the vehicle in the second path. When a vehicle moves from a path to another, the lane-merging activity happens. Fig. 11(c) shows a result. For scene S2, when an object enters the scene, it is classified into vehicle or pedestrian. For each vehicle class, we have already learned the primary motion patterns for each block. For a trajectory, if its probability $P(T|g_m^*)$ is smaller than th_{m^*}, we consider it as an abnormal activity. To make a quantitative results, we randomly select 100 trajectories from traffic scene S2. These trajectories include 21 abnormal trajectories (class 1) and 79 normal trajectories (class 2). Recall and Precision are adopted to measure the performance. We compare our model with the learned model by method II introduced in Section V-B2. For the baseline method denoted as Ma, we calculate the average distance between two trajectories in each cluster and use this distance as a threshold to decide whether a trajectory is abnormal or not. The results are shown in Table V and demonstrate our method Mb performs better. An example is shown in Fig. 11(d).

VI. DISCUSSIONS AND CONCLUSION

In this paper, a novel framework is proposed to mine semantic context information for intelligent video surveillance of traffic

scenes. First, we introduce how to learn scene-specific context information from object-specific context information. Then, object classification is improved by combining of multiple features under a cotraining framework. Based on the learned information, we adopt it to improve object detection and tracking, and detect abnormal events. Experimental results validate that the semantic context information is effective to improve object detection, object classification, object tracking and abnormal event detection. For object detection, our results in Section V-D show the proposed method is effective for far-field and medium-field surveillance videos, and obtain bad results for near-field videos. For object classification, due to the cotrained framework, our method can fuse the strength of different features and increase the training samples by the semi-supervised learning method automatically. For the six testing traffic scenes, there are existing different viewing angles, shadows, low resolution, illumination changes and different environmental effects. However, compared with the previous methods, such as the AdaBoost classifier [19], the LDA-based classifier and LLC classifier [49], our cotrained classifier obtains the best results and has about more than 9% improvement in several traffic scenes. Due to considering the scene-specific context information, object tracking is improved as shown in Section V-E. Moreover, our system can also detect the abnormal event that vehicles break traffic rules as shown in Section V-F. In the future, we will investigate how to use scene-specific context information to improve pedestrian detection and abnormal event detection in more complex conditions.

REFERENCES

[1] O. Javed, K. Shafique, and M. Shah, "Automated visual surveillance in realistic scenarios," *IEEE Multimedia*, vol. 14, no. 1, pp. 30–39, Jan. –Mar. 2007.

[2] R. T. Collins, A. J. Lipton, and T. Kanade, "Introduction to the special section on video surveillance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 745–746, Aug. 2000.

[3] T. Cucinotta, L. Palopoli, L. Abeni, D. Faggioli, and G. Lipari, "On the integration of application level and resource level qos control for real-time applications," *IEEE Trans. Ind. Inf.*, vol. 57, no. 4, pp. 479–491, Nov. 2010.

[4] E. Camponogara, A. de Oliveira, and G. Lima, "Optimization-based dynamic reconfiguration of real-time schedulers with support for stochastic processor consumption," *IEEE Trans. Ind. Inf.*, vol. 57, no. 4, pp. 594–609, Nov. 2010.

[5] G. Wang, L. Tao, H. Di, X. Ye, and Y. Shi, "A scalable distributed architecture for intelligent vision system," *IEEE Trans. Ind. Inf.*, vol. 8, no. 1, pp. 91–99, Feb. 2012.

[6] C. Stauffer and E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[7] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1450–1464, Sep. 2006.

[8] I. Saleemi, K. Shafique, and M. Shah, "Probabilistic modeling of scene dynamics for applications in visual surveillance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1472–1485, Aug. 2009.

[9] T. Bcher, C. Curio, J. Edelbrunner, C. Igel, D. Kastrup, I. Leefken, G. Lorenz, A. Steinhage, and W. V. Seelen, "Image processing and behaviour planning for intelligent vehicles," *IEEE Trans. Ind. Electron.*, vol. 90, no. 50, pt. 1, pp. 62–75, Feb. 2003.

[10] J. C. Nascimento and J. S. Marques, "Performance evaluation for object detection algorithms for video surveillance," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 761–774, Oct.–Dec. 2006.

[11] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. IEEE Frame-Rate Workshop*, 2000.

[12] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using non-parametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.

[13] S. Chen, J. Zhang, Y. Li, and J. Zhang, "A hierarchical model incorporating segmented regions and pixel descriptors for video background subtraction," *IEEE Trans. Ind. Inf.*, vol. 8, no. 1, pp. 118–127, Feb. 2012.

[14] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[15] T. Zhao and R. Nevatia, "Bayesian human segmentation in crowded situations," in *Proc. CVPR*, 2003.

[16] L. Dong, V. Parameswaran, V. Ramesh, and I. Zoghlami, "Fast crowd segmentation using shape indexing," in *Proc. ICCV*, 2007.

[17] M. Kafai and B. Bhanu, "Dynamic Bayesian networks for vehicle classification in video," *IEEE Trans. Ind. Inf.*, vol. 8, no. 1, pp. 100–109, Feb. 2012.

[18] R. Cutler and L. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 781–796, Aug. 2000.

[19] L. Zhang, S. Li, X. Yuan, and S. Xiang, "Real-time object classification in video surveillance based on appearance learning," in *Proc. IEEE Int. Workshop Visual Surveillance in Conjunction With CVPR*, 2007.

[20] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with cotraining," in *Proc. 11th Annu. Conf. Computational Learning Theory*, 1998.

[21] A. Levin, P. Viola, and Y. Freund, "Unsupervised improvement of visual detectors using co-training," in *Proc. ICCV*, 2003.

[22] P. Vadakkepat, P. Lim, L. C. D. Silva, L. Jing, and L. L. Ling, "Multimodal approach to human-face detection and tracking," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1385–1393, Jun. 2008.

[23] B.-F. Wu, C.-T. Lin, and Y.-L. Chen, "Dynamic calibration and occlusion handling algorithms for lane tracking," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1757–1773, Oct. 2009.

[24] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram, "Human activity recognition using multidimensional indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1091–1104, Aug. 2002.

[25] C. Tran and M. M. Trivedi, "3-d posture and gesture recognition for interactivity in smart spaces," *IEEE Trans. Ind. Inf.*, vol. 8, no. 1, pp. 178–187, Feb. 2012.

[26] X. Wang, X. Ma, and E. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 539–555, Mar. 2009.

[27] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[28] D. Herrero-Perez and H. Martinez-Barbera, "Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems," *IEEE Trans. Ind. Inf.*, vol. 6, no. 2, pp. 166–180, May 2010.

[29] X. Wang, K. T. Ma, G.-W. Ng, and E. Grimson, "Trajectory analysis and semantic region modeling using a nonparametric Bayesian model," in *Proc. CVPR*, 2008.

[30] Y. Yang, J. Liu, and M. Shah, "Video scene understanding using multi-scale analysis," in *Proc. ICCV*, 2009.

[31] I. Saleemi, L. Hartung, and M. Shah, "Scene understanding by statistical modeling of motion patterns," in *Proc. CVPR*, 2010.

[32] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2287–2301, Nov. 2011.

[33] D. Makris and T. Ellis, "Automatic learning of an activity-based semantic scene model," in *Proc. IEEE Conf. Adv. Video Signal Based Surveillance*, 2003.

[34] X. Wang, K. Tieu, and E. Grimson, "Learning semantic scene models by trajectory analysis," in *Proc. ECCV*, 2006.

[35] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and Maybank, "A system for learning statistical motion patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1450–1464, Sep. 2006.

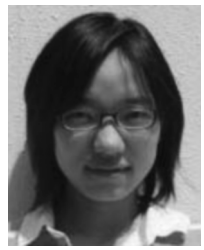
[36] S. H. and C. S. , "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Jan. 1978.

[37] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[38] T. Zhang, H. Lu, and S. Li, "Learning semantic scene models by object classification and trajectory clustering," in *Proc. CVPR*, 2009.
[39] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 1222–1239, Aug. 2001.
[40] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 147–159, Jan. 2004.
[41] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.
[42] T. Zhang, S. Z. Li, S. Xiang, L. Zhang, and S. Liu, "Co-training based segmentation of merged moving objects," *Proc. Video Surveillance*, 2008.
[43] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognit.*, pp. 51–59, 1996.
[44] P. A. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.
[45] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Statistics*, 2000.
[46] J. Rittscher, P. Tu, and N. Krahnstoever, "Simultaneous estimation of segmentation and shape," in *Proc. CVPR*, 2005.
[47] i-lids dataset for avss, 2007 [Online]. Available: http://www.avss2007.org
[48] M. Meila and J. Shi, "A random walks view of spectral segmentation," *AI and Statistics (AISTATS)*, 2001.
[49] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. CVPR*, 2010.
[50] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005.
[51] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. CVPR*, 2006.

Tianzhu Zhang (M'11) received the B.S. degree in communications and information technology from Beijing Institute of Technology, Beijing, China, in 2006, and the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2011.

He is currently with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, as well as with the China-Singapore Institute of Digital Media, Singapore. Prior to this, he was a Postdoctoral Fellow with the Advanced Digital Sciences Center (ADSC), a joint research center between the University of Illinois at Urbana-Champaign (UIUC) and the Agency for Science, Technology and Research (A*STAR), Singapore. He does extensive research on computer vision and multimedia, such as action recognition, video surveillance, and object tracking.



Si Liu (S'12) is currently working toward the Ph.D. degree at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

She is currently a Research Assistant with the Learning and Vision Group, National University of Singapore. Her research interests includes computer vision and multimedia.

Changsheng Xu (M'97–SM'99) is a Professor with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and Executive Director of China-Singapore Institute of Digital Media. His research interests include multimedia content analysis/indexing/retrieval, pattern recognition and computer vision. He has 30 granted/pending patents and has authored and coauthored over 200 refereed research papers in these areas. He is an associate editor of the *ACM Transactions on Multimedia Computing, Communications and Applications* and *Multimedia Systems Journal*.

Dr. Xu served as Program Chair of ACM Multimedia 2009. He has served as associate editor, guest editor, general chair, program chair, area/track chair, special session organizer, session chair and TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences and workshops.

Hanqing Lu (SM'06) received the B.E. and M.E. degrees from Harbin Institute of Technology, Harbin, China, in 1982 and 1985, respectively, and the Ph.D. degree from Huazhong University of Sciences and Technology, Wuhan, China, in 1992.

Currently, he is a Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include image similarity measure, video analysis, object recognition and tracking. He authored and coauthored more than 200 papers in those areas.