Local Fast R-CNN Flow for Object-centric Event Recognition in Complex Traffic Scenes

Qin Gu[1,2], Jianyu Yang[1,], Wei Qi Yan[2], Yanqiang Li, and Reinhard Klette[2]

[1] University of Electronic Science and Technology of China, Chengdu
Sichuan 611731, P.R. China, guqin.uestc@outlook.com
[2] Auckland University of Technology, Auckland, 1010, New Zealand
[3] Shandong Provincial Key Laboratory of Automotive Electronics and Technology
Jinan 250000, P.R. China

Abstract. This paper presents a solution for an integrated object-centric event recognition problem for intelligent traffic supervision. We propose a novel event-recognition framework using deep local flow in a fast region-based convolutional neural network (R-CNN). First, we use a fine-tuned fast R-CNN to accurately extract multi-scale targets in the open environment. Each detected object corresponds to an event candidate. Second, a deep belief propagation method is proposed for the calculation of local fast R-CNN flow (LFRCF) between local convolutional feature matrices of two non-adjacent frames in a sequence. Third, by using the LFRCF features, we can easily identify the moving pattern of each extracted object and formulate a conclusive description of each event candidate. The contribution of this paper is to propose an optimized framework for accurate event recognition. We verify the accuracy of multi-scale object detection and behavior recognition in extensive experiments on real complex road-intersection surveillance videos.

Keywords: deep learning, event recognition, convolutional neural network, belief propagation

1 Introduction

Object-centric event recognition is pivotal for traffic violation recording, traffic monitoring, and traffic control [9]. Vision-based intelligent transportation surveillance systems have been an active research area in past decades due to high credibility and low costs of those systems.

There are various definitions for an *event*. In general, an event in video content refers to an object of interest with a certain behavior in the given scenes. Here, we focus on traffic scenes at road intersections; our object-centric events include object detection, object recognition, and object behavior recognition in an interval of time.

Representing different object-centric events usually leads to high computational costs because a single event requires (in general) object motion detection, object tracking, and object behavior understanding.

Following those three steps, robust object detection is the key step for event modeling. Low-level feature-based methods such as Gabor wavelets, histogram of gradients (HOG), or optical flow have been used in pioneering research in this field. Recently, deep learning [14] achieves remarkable advances in solving the given problems. Deep learning defines the state-of-the-art approach for object detection or human activity recognition. However, deep learning is also run-time consuming when detecting, tracking, and understanding objects, class by class. Taking the need for real-time traffic monitoring and analysis into consideration, we concluded that traditional scanning of all potential patches is impractical.

In this paper, we combine moving object detection as well as tracking and event recognition with a *convolutional neural network* (CNN) using a *local fast region-based CNN flow* (LFRCF) descriptor. For this purpose, the already well-studied *fast region-based CNN* (fast R-CNN) architecture is fine-tuned for fast event candidate generation. Next, spatio-temporal motion information is compressed into local region flow in the deep convolutional space for event representation, called *deep local flow*. Finally, the LFRCF is used for further event recognition and identification. The contributions in this paper are as follows:

1. A novel LFRCF descriptor is proposed using deep belief propagation.
2. We propose a fine-tuned fast R-CNN architecture for automatically generating a group of regions of interest for real-time traffic event recognition.
3. We investigate a particular framework of deep CNNs, trained for integrated object detection and behavior recognition in video data.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 details the proposed event recognition method using our LFRCF descriptor. Section 4 shows experimental results for verifying the proposed method. Section 5 concludes.

2 Related Work

In general, object-centric event recognition algorithms follow three steps, briefly outlined below.

First, selective object detection and motion estimation can be beneficial for both speed-up and accuracy. A Gaussian mixture model (GMM) is used in [26] for vehicle detection in complex urban traffic scenes.

A diversity of feature-vector representation schemes has been proposed for object detection in complex scenes. The active basis model [24] has been widely employed for vehicle detection [11,16] in traffic surveillance. With the assistance of a shared skeleton method, it can be easily trained with a considerable detection performance. However, it can only be used for one object with a fixed pose. An AND-OR graph [15] has been proposed for vehicle detection in congested traffic conditions. A deformable part-based model for object detection was introduced in [5]. These two methods still need multi-models for various targets and multiple viewing points, which is rather time-consuming.

Deep learning methods improved dramatically the state-of-the-art in visual object detection and recognition. The CNN [13] powered the performance of object detection and recognition. Recently, a focus in this area [6,20] is on process acceleration with a fundamental algorithm for region search [22]. However, it is still a challenge to detect, track, and analyse the behavior for moving targets in continuous frames, even with GPU-enabled computing.

Second, object tracking algorithms are proposed for trajectory reconstruction. Region-based tracking algorithms [8], feature-based tracking algorithms [25], and model-based tracking algorithms [19] have all been widely applied for various outputs.

Third, for the tracking of moving objects in adjacent frames of a video sequence, the problem of understanding object behaviors from image sequences arises naturally. Subsequently, methods such as hidden Markov models (HMM) [2], Bayesian approaches [3], or 3-dimensional (3D) models [10] are used to understand the trajectory of moving targets.

There is also work [17, 21] that aims at a more focused anomaly detection, but so far in a global sense only, not for individually acting objects. Global anomalies (involving multiple objects) are, for example, a traffic jam, an accident, or changes of global motion in scenes of crowds. Global anomaly detection also requires further research.

Different to existing work, our contribution in this paper is an integrated framework for multi-class event recognition in complex road scenes. Event localization and recognition are conducted in a deep CNN using the proposed LFRCF (i.e. local fast R-CNN flow) descriptor.

3 Methodology

This section presents our event recognition method using a fast R-CNN [6] architecture and the proposed LFRCF descriptor. We divide this section into three parts. First, we provide an overview of the event recognition framework. Then we present the fundamental method of fast R-CNN. Finally, we detail the calculation of the LFRCF descriptor with deep belief propagation for behavior recognition and event identification.

3.1 Overview

As illustrated in Fig. 1, the proposed framework recognizes a multi-scale and object-centric event by using two non-adjacent frames, denoted as Frame `input1` and Frame `input2`. First, fast R-CNN is implemented for convolutional feature extraction, object detection, and bounding box regression. Hence, by using the extracted location of each bounding box, we use a new spatial-temporal pooling algorithm to extract the local convolutional feature (i.e. a local Conv feature map) in the 4th convolutional layer, for two non-adjacent frames. Finally, the LFRCF descriptor is calculated between the obtained two local Conv feature maps, which refers to the moving patterns of candidate events; this descriptor is the applied for the final behavior recognition and event identification.

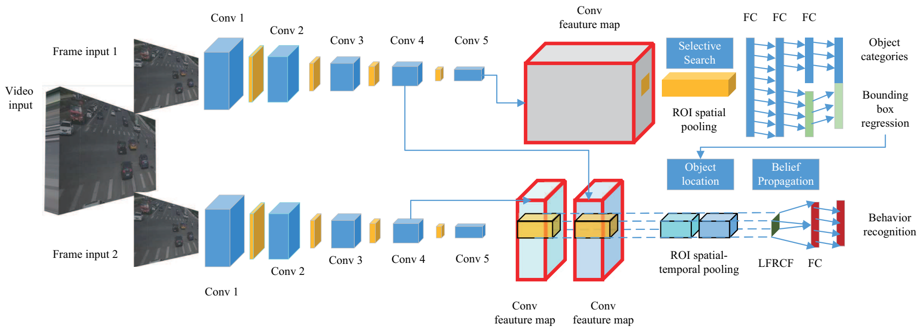

Fig. 1. Framework of the proposed method

3.2 Region-based Convolutional Neural Network

In this paper, we use convolutional layers, max pooling layers, rectified linear units (ReLUs), and fully connected (FC) layers to construct our traffic-event recognition network.

Input data pass through all the organized layers to generate the final recognition outputs. In the convolution layers, a group of kernels is used to filter the input such as to produce feature maps for deeper feature extraction. The function of the pooling layer is to calculate the overall response of a neighborhood area in a feature map, which is one of the outputs of the convolution layer. Being aware of the problem of over-fitting, dropout layers are proposed for training towards optimization. Finally, by using the softmax optimization method, a multi-class identification result is given with an FC layer.

One of the most effective approaches for solving a multi-scale object detection task is the family of R-CNNs. By using the selective search algorithm, a group of regions is extracted for further scale normalization (i.e. resized to 227×227 in our case).Then, a traditional CNN is used for feature extraction and object recognition. This method achieves high accuracy, however, it is very time consuming because of redundant feature extraction in a deep ConvNet.

In this paper, we use a fast R-CNN framework for multi-scale object detection and event hypothesis generation. This framework solves the previous problems by computing the feature map only once per image. The corresponding region-based deep convolutional feature map is extracted in form of a new ROI-pooling map. We also propose a multi-task loss function for bounding-box regression.

The ConvNet is regarded as a feature descriptor of specific video frames. In this paper, we use the *VGG-16* deep ConvNet for our fast R-CNN, which is pre-trained on a large *Pascal Voc 2007* dataset, as Table 1 shows.

Categories and bounding box locations of 20 classes of objects are used to finish the training stage.

By using a multi-task loss-function training algorithm, the loss (or error) function during the fine-tuning work of the initial VGG-16 neural network is

Table 1. Details of all selected layers of the pre-trained fast R-CNN

Layer	conv1	pool1	conv2	pool2	conv3	pool3	conv4	pool4	conv5
Input	600	600	300	300	150	150	75	75	38
	×975	×975	×488	×488	×244	×244	×122	×122	×61
	×3	×64	×64	×128	×128	×256	×256	×512	×512
Output	600	300	300	150	150	75	75	38	38
	×975	×488	×488	×244	×244	×122	7×122	×61	×61
	×64	×64	×128	×128	×256	×256	×512	×512	×512
Channel	64	64	128	128	256	256	512	512	512

represented as follows:

$$E(\mathcal{X}, \mathcal{C}, \mathcal{L}) = E_{cls}(f(\mathcal{X}), \mathcal{C}) + \zeta \cdot E_{loc}(\mathcal{X}, \mathcal{L}) \tag{1}$$

where, $\mathcal{X}$ is the location of the considered region of interest (ROI) defined by (r_1, r_2, c_1, c_2). $\mathcal{C}$ and $\mathcal{L}$ are the ground truth for object category and location, respectively. $E_{cls}(f(\mathcal{X}), \mathcal{C})$ is the loss function regarding the recognition of an object in $\mathcal{X}$ as being in the correct class $\mathcal{C}$. The second term E_{loc} is the loss function of bounding box regression; ζ is a parameter to control the balance between these two terms. In this paper, we simply use $\zeta = 1$.

3.3 Local fast R-CNN Flow

Accurate object detection and recognition results define already an important step towards accurate event analysis. However, in order to give a detailed description of an object-centric event for recognition, we still need to analyze the motion pattern of any detected and recognized object of interest.

Flow is an effective approach for moving pattern description. However, pixelwise flow matching in a large-scaled spatial image is time-consuming and inaccurate. Therefore, we directly analyze the flow in the local region of interest as fast R-CNN flow, or *deep flow* for short.

Feature reorganization. For extracting flow features in the extracted bounding box by using fast R-CNN, we reorganize convolution features in extended bounding boxes in two non-adjacent frames.

The convolutional feature map in the k-th layer is taken into account, formally expressed by

$$f_c(V(t), k) = C_k^t \in \mathbb{R}^{h_k \times w_k \times d_k}, \text{ for } k = 1, \ldots, 5 \tag{2}$$

Here, $V(t)$ is the t-th frame in video V, and C_k^t is the feature map in the k-th convolutional layer of the t-th frame.

By implementing this method of feature description for two selected non-adjacent frames (e.g. $t - a$ and $t + b$), we obtain two convolutional feature maps C_k^{t-a} and C_k^{t+b}; k is the index of the convolutional layers. These two maps indicate the motion patterns within the interval of $a + b + 1$ frames.

The location of recognized Object n in Frame $t-a$ is expressed by

$$\mathcal{X}_f(n) = [r_1(n), r_2(n), c_1(n), c_2(n)] \tag{3}$$

where r_1, r_2, c_1, c_2 are the row and column coordinates of the object's bounding box in this frame. Considering that an object may easily move out of its current bounding box in a short period of time, we construct a spatial extended bounding box for each object. Let the standard extending scale s_t be defined as follows:

$$s_t(n) = \eta \cdot \max\left(\left\lfloor \frac{r_2(n) - r_1(n)}{2} \right\rfloor, \left\lfloor \frac{c_2(n) - c_1(n)}{2} \right\rfloor\right) \tag{4}$$

where η is a parameter to change the scale. Thus, the n^{th} extended region is represented as

$$\begin{aligned} \mathcal{X}_e(n) = [&\max(1, r_1(n) - s_t(n)), \min(M, r_2(n) + s_t(n)), \\ &\max(1, c_1(n) - s_t(n)), \min(N, c_2(n) + s_t(n))] \end{aligned} \tag{5}$$

M and N are the row and column numbers of the whole image.

By using the spatial extended bounding box region, we further reorganize the k^{th} convolutional feature matrix for object-centric behavior recognition. The reorganized features, for considered object n in two non-adjacent frames, are further written as two local feature matrices $L_k^{t-a,n}$ and $L_k^{t+b,n}$. $L_k^{t-a,n}$ and $L_k^{t+b,n}$ consist of all pixels in the tensor $C_k^{t-a}(x,y,z)$ and $C_k^{t+b}(x,y,z)$, with

$$\lceil \chi_1 \cdot \mathcal{X}_e(1,1) \rceil \leq x \leq \lfloor \chi_1 \cdot \mathcal{X}_\rceil(1,2) \rfloor \tag{6}$$

$$\lceil \chi_2 \cdot \mathcal{X}_e(1,3) \rceil \leq y \leq \lfloor \chi_2 \cdot \mathcal{X}_e(1,4) \rfloor \tag{7}$$

for $1 \leq z \leq d_k$. $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ are the ceiling and floor function, respectively. χ_1 and χ_2 are the layer's scale factors, with

$$\chi_1 = \frac{h_k}{M}, \;\; \chi_2 = \frac{w_k}{N} \tag{8}$$

After feature reorganization, we extract the local convolution feature matrix $L_k^{t-a,n}$ and $L_k^{t+b,n}$ for each object which has already been recognized by the fast R-CNN framework. These matrices are used for recognizing events occurring between the selected two non-adjacent frames.

LFRCF matching. Following related work [7], we transfer the behavior recognition problem into a label parsing problem using probabilistic graphical model. Two local convolution feature matrices $L_k^{t-a,n}$ and $L_k^{t+b,n}$ are used to analyze the behavior of the n^{th} object which is detected in Frame $t-a$. For convenience, we directly set the value of a and b as 0 and 5, respectively, in this paper.

Let $w(p) = [u(p), v(p)]$ be the LFRCF at a pixel location $p \in \Omega$, where Ω is the set of all pixel locations of a map $L_k^{t-a,n}$. To calculate the LFRCF w from

$L_k^{t-a,n}$ to $L_k^{t+b,n}$, we introduce a unary *cost function* $E(w)$ as follows:

$$\begin{aligned} E(w) = & \sum_{p \in \Omega} \min(\left\| {L_k}^{t+b}(p) \right\|_1 - \left\| {L_k}^{t-a}(p + w(p)) \right\|_1, d) \\ & + \sum_{q \in A(p)} \{\min(\alpha \left| u(p) - u(q) \right|, e) \\ & + \min(\alpha \left| v(p) - v(q) \right|, e)\} \end{aligned} \tag{9}$$

Here, $A(p)$ is the set of pixel locations being 4-adjacent with p; d and e are two thresholds for truncating the L_1 norms.

We have two terms in the cost function, namely the data and the smoothness term. By using the data term, the convolutional event descriptor is constrained to be matched; the smoothness term is employed to constrain adjacent pixels to having similar LFRCFs.

For minimizing the cost function $E(w)$ and obtaining the most accurate directional LFRCF vector w, we use belief propagation (BP) [18]. Compared with the method in [18], the proposed cost function ignores the small-displacement term, as the object may move obviously between non-adjacent frames.

By using an improved loopy belief propagation algorithm, the cost function is minimized after 40 iterations for each object. In this paper, due to the next two reasons, instead of using the calculated LFRCF in the area $\mathcal{X}_e(n)$, we retrospect the location $\mathcal{X}_f(n)$ and extract the central LFRCF for further behavior representation.

First, we only focus on the moving pattern of an object which has already been located in the area $\mathcal{X}_f(n)$, which corresponds to the central area of $\mathcal{X}_e(n)$. Second, by considering the convolutional processing in the neural network we may influence the boundary of on object in an image.

Let $m_l(n)$, $n_l(n)$ be the height and width of map $L_k^{t-a,n}$, respectively. Thus, the central LFRCF area for further behavior representation is given by

$$[\eta \cdot \max(m_l, n_l), m_{l-}\eta \cdot \max(m_l, n_l), \eta \cdot \max(m_l, n_l), n_{l-}\eta \cdot \max(m_l, n_l)] \tag{10}$$

The behavior recognition result is denoted by $w(x_l, y_l)$, with $\eta \cdot \max(m_l, n_l) \leq x_l \leq m_{l-}\eta \cdot \max(m_l, n_l)$, and $\eta \cdot \max(m_l, n_l) \leq y_l \leq n_{l-}\eta \cdot \max(m_l, n_l)$.

Motion visualization. The parsing result, which corresponds to dynamic local motion, is represented as shown in Fig. 2 by using a color-key. Then, based on the visualization results of the proposed LFRCF descriptor, the behavior can be easily distinguished. In order to further clarify an object-centric event in complex on-road scenes, we further provide a motion-representation method for events.

A behavior vector is calculated as follows:

$$\begin{aligned} u_m = \frac{1}{Z} \sum_{x,y} u(x, y), \quad & v_m = \frac{1}{Z} \sum_{x,y} v(x, y) \\ x \in [\eta \cdot \max(m_l, n_l), & m_{l-}\eta \cdot \max(m_l, n_l)] \\ y \in [\eta \cdot \max(m_l, n_l), & n_{l-}\eta \cdot \max(m_l, n_l)] \end{aligned} \tag{11}$$

Calculated u_m and v_m are the mean horizontal and vertical flow of the calculated LFRCF, u and v are the horizontal and vertical LFRCF flow components, respectively. Z is the area of the current region. See Fig. 3 for examples of event representation.

In Fig. 4, the green rectangular lines are regression results of object detection and event location. The directed red bar in the blue circle represents the central mean flow $w_m = [u_m, v_m]$. It accurately represents the behavior of the detected object, and even provides a coarse information of the current speed of the object.

4 Experiments

The experimental report is divided into three segments. Detailed information of the dataset is given at the beginning. Then, we compare the performance of event localization for different methods. Finally, we present the performance of event recognition for extensive data recorded at a real traffic intersection using the proposed method.

Datasets. It is always a challenging problem to detect and track vehicles and pedestrians in outdoor scenes for traffic event recognition. In this paper, focusing on various scenes, we use three groups of datasets which are selected from the publicly available *HIGHWAY* dataset and the *UA-DETRAC* dataset. These video datasets are captured under various lighting conditions, viewing angles, and for different road scenes.

To evaluate the proposed method, we also collected an extensive data-set, called the *JINAN* data-set, at inner-city road intersections with a camera located about 8 meters over the road surface. Our videos record top or rear views of vehicles moving below the camera level. It is possible to observe in the recorded data vehicles and pedestrians in a distance such as on the other side of the intersection. These videos were recorded at a frequency of 25 frames per second.

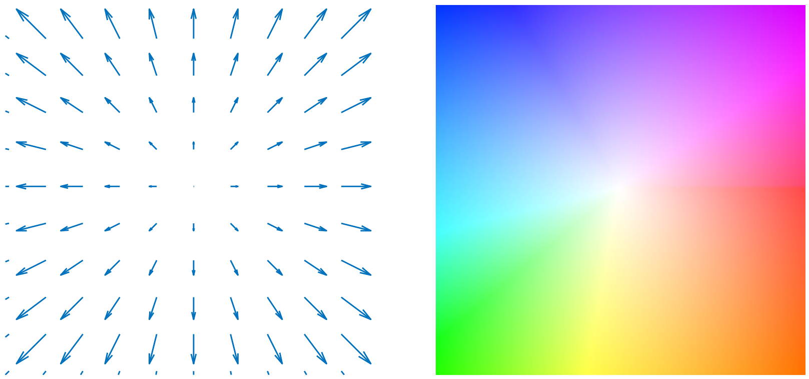

Fig. 2. Visualization of deep convolution flow. *Left*: Flow directions and magnitudes for selected pixels. *Right*: Color-key-based representation of optical flow for all pixels

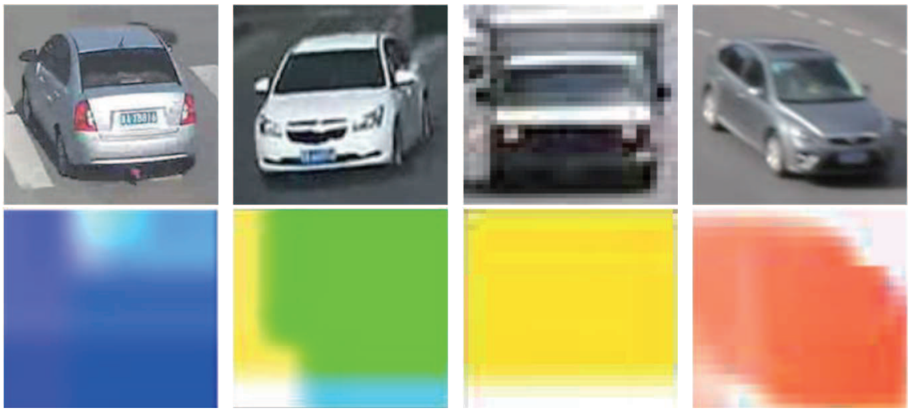

Fig. 3. LFRCF visualization for multi-scale events

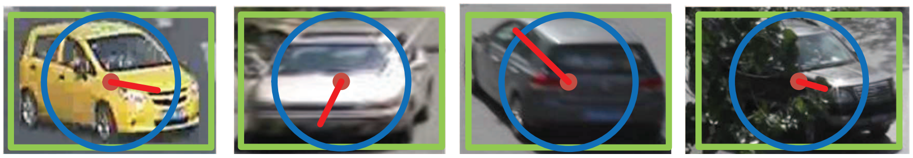

Fig. 4. Event localization and recognition examples.

Table 2. Used datasets for object detection and tracking, and event characterisation

	Resolution	Frames	Frames showing vehicle events	Frames showing pedestrian events
HIGHWAY	640 × 480	1,652	742	0
UA-DETRAC	720 × 960	1,466	426	0
JINAN	2,592 × 2,048	1,304	179	97

Summarizing ground truth for event localization, brief event descriptions (for three selected time intervals of traffic videos) are listed in Table 2.

Event Localization. Normally, robust object detection and tracking are necessary for event localization. In order to identify object movements, we compare the performance of vehicle localization of the *active basis model* of [11], the *Viola-Jones cascade detector* of [23], the *deformable part model* of [4], and our proposed method.

Table 3. Comparisons of event localization on the *HIGHWAY* data-set

	Active basis model [11]	V-J cascade detector [23]	Deformable part model [4]	Proposed method
Recall	83.2%	94.5%	46.1%	92.7%
Precise	56.1%	45.3%	91.5%	98.4%

Table 4. Comparisons of event localization on the *UA-DETRAC* data-set

	Active basis model [11]	V-J cascade detector [23]	Deformable part model [4]	Proposed method
Recall	77.3%	84.8%	90.3%	88.7%
Precise	84.2%	50.6%	91.9%	97.1%

Table 5. Comparisons of event localization on the *JINAN* data-set

	Active basis model [11]	V-J cascade detector [23]	Deformable part model [4]	Proposed method
Recall	-	-	74.8%	81.8%
Precise	-	-	97.8%	94.3%

The active basis model and the V-J cascade detector perform well for rear-view vehicles, but it is difficult to train them for accurate vehicle event localization in cases of other viewing-angles (see the *JINAN* data-set results). The deformable part-based model is very accurate for detecting objects with a rigid structure, but it costs too much time to process one frame for one kind of targets even when using a cascading speed-up technology. Besides, we even need to cope with whole frames several times to extract different objects. The method proposed by us shows competitive results but proves to be much more time-efficient for extracting multiple moving objects of interest. A further validation of recognition accuracy is given in the next section.

Table 6. Comparison of comprehensive event recognition results

	Accuracy	Computational cost
Optic flow [1]	63.1%	0.9 s (CPU)
Dense SIFT flow [18]	98.9%	10.9 s (CPU)
Proposed method	98.7%	0.4 s (CPU)

Event Recognition. In this paper, multi-class object-centric (i.e. vehicle and pedestrian) events are taken into consideration. For an event sample, we selected five subsequent frames. The manually labeled 1,347 vehicle-centric and 97 pedestrian-centric events are studied for recognition by the proposed method.

According to the calculated behavior vector $[u_m, v_m]$, we generally define nine directions of motion (i.e. bottom-up, up-bottom, right-left, left-right, bottomright-upleft, bottomleft-upright, upright-bottomleft, upleft-bottomright, and remaining static). Based on the manually labeled result, accurate event recognition is defined by matching flow direction and flow magnitude.

The accuracy A is defined as follows:

$$A = \frac{P_p + P_v}{N_p + N_v} \tag{12}$$

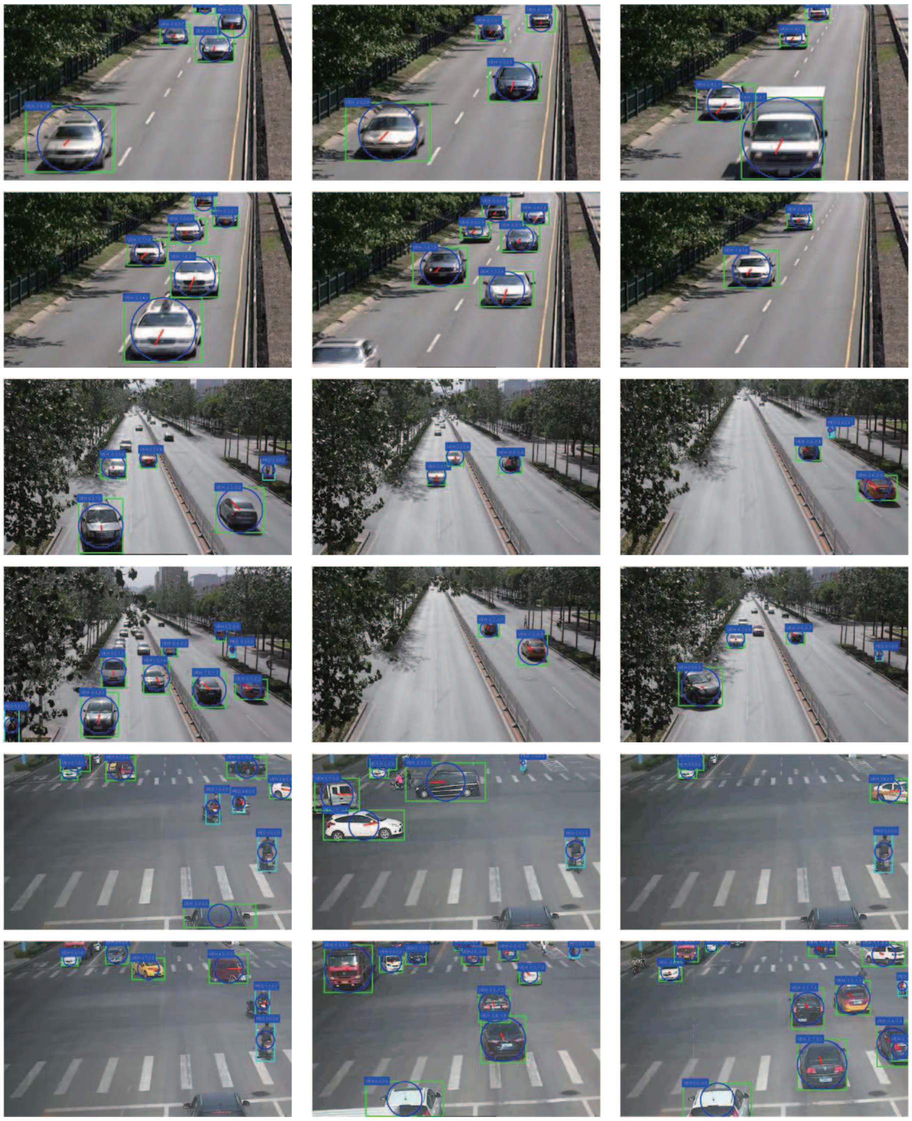

Fig. 5. Event recognition performance. *Top to bottom*: Results for the *HIGHWAY*, *UA-DETRAC*, and *JINAN* data-sets

where P_p and P_v are the number of accurately recognized events for pedestrians and vehicles, respectively. N_p and N_v are the number of all manually labeled events for pedestrians and vehicles.

We also use optic flow and dense SIFT flow for comparison on behavior recognition; see Table 6. Here, the 1st and the 5th frame are used for deep flow calculation.

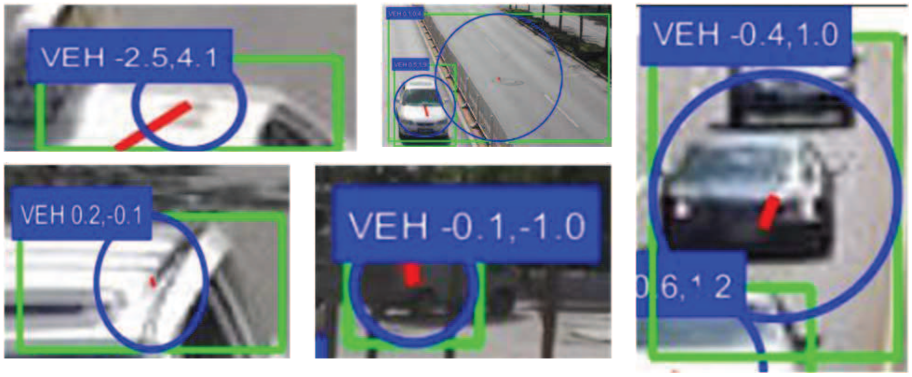

Fig. 6. Examples of false event recognition

By using the proposed event recognition framework, each moving object of interest is detected, frame by frame. The entire algorithm is implemented in Matlab 2016a with *MatConvNet* in OS Windows 10, using 16GB RAM and an i5 CPU processor. Processing is on average at 0.2 fps on a CPU-only mode. Some false event recognition examples (i.e. false-positive detection and false-flow calculation) are given in Fig. 6 illustrating a need for further refinements.

5 Conclusions

This paper presents a novel traffic-event recognition method using a descriptor defined by local fast R-CNN flow (LFRCF). Specifically, we use a fine-tuned fast R-CNN for multi-scaled object (i.e. vehicles or pedestrians) detection in complex traffic scenes. Then, by using a new spatial-temporal pooling algorithm, we extract the proposed LFRCF descriptor in the 4th convolutional layer for each object. Local convolutional features of two non-adjacent frames are used for event recognition with an improved loopy belief propagation algorithm.

By using the identified three datasets (i.e. data collected under various lighting, viewing angles, and for different road scenes), we evaluate the robustness and accuracy of traffic-event recognition and the computing cost of the proposed method.

It might be of interest to extend DCF analysis to the recognition of more generalized events. A high-level understanding (e.g. by linking multiple traffic events of the same vehicle together) is also worth considering.

Acknowledgement. The experimental work was partially supported by Shandong Provincial Key Laboratory of Automotive Electronics and Technology, Institute of Automation, Shandong Academy of Sciences.

References

1. Baker, S., Scharstein, D., Lewis, J.P., and Szeliski,R.: A database and evaluation methodology for optical flow. Int. J. Computer Vision, 92(1), 1–31 (2011)
2. Bashir, F.I., Khokhar, A.A., and Schonfeld, D.: Object trajectory-based activity classification and recognition using hidden Markov models. IEEE Trans. Image Processing, 16(7), 1912–1919 (2007)
3. Dore, A. and Regazzoni, C.: Interaction analysis with a Bayesian trajectory model. IEEE Trans. Intelligent Systems, 16(7), 1912–1919 (2007)
4. Felzenszwalb, P.F., Girshick, R.B., and McAllester, D.: Cascade object detection with deformable part models. Proc. IEEE Conf. Computer Vision Pattern Recognition, pages 2241–2248 (2010)
5. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., and Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Analysis Machine Intelligence, 32(9), 1627–1645 (2010)
6. Girshick, R.: Fast R-CNN. Proc. IEEE Int. Conf. Computer Vision, pages 1440–1448 (2015)
7. Gu, Q., Yang, J., Cui, G.,Ling, K., Hua, Z., and Klette, R.: Multi-scale vehicle logo recognition by directional dense SIFT flow parsing. Proc. IEEE Int. Conf. Image Processing, pages 3827–3831 (2016)
8. Gupte, S.O., Masoud, O., Martin, R.F.K, and Papanikolopoulos, N.P.: Detection and classification of vehicles. IEEE Trans. Intelligent Transportation Systems, 3(1), 37–47 (2002)
9. Hu, W., Tan, T., Wang, L., and Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Trans. Systems Man Cybernetics, Part C, 34(3), 334–352 (2004)
10. Hu, W., Xiao, X., Xie, D., Tan, T., and Maybank, S.: Traffic accident prediction using 3D model-based vehicle tracking. IEEE Trans. Vehicular Technology, 53(3), 677–694 (2004)
11. Kamkar, S. and Safabakhsh, R.: Vehicle detection, counting and classification in various conditions. IET Intelligent Transport Systems, 10(6), 406–413 (2016)
12. Klette, R.: Concise Computer Vision. Springer, London (2014)
13. Krizhevsky, A., Sutskever, I., and Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Proc. Advances Neural Information Processing Systems, pages 1097–1105 (2012)
14. LeCun, Y., Bengio,Y., and Hinton, G.: Deep learning. Nature, 521(7553), 436–444 (2015)
15. Li, Y., Li, B., Tian, B., and Yao, Q.: Vehicle detection based on the and–or graph for congested traffic conditions. IEEE Trans. Intelligent Transportation Systems, 14(2), 984–993 (2013)
16. Li, Y., Li, B., Tian, B., and Yao, Q.: Vehicle detection based on the deformable hybrid image template. Proc. IEEE Int. Conf. Vehicular Electronics Safety, pages 114–118 (2013)
17. Li, Y., Liu, W., and Huang, Q.: Traffic anomaly detection based on image descriptor in videos. Multimedia Tools Applications, 75(5), 2487–2505 (2016)
18. Liu, C., Yuen, J., Torralba, A.: SIFT flow: Dense correspondence across scenes and its applications. IEEE Trans. Pattern Analysis Machine Intelligence, 33(5), 978–994 (2011)
19. Niknejad, H.T., Takeuchi, A., Mita, S., and McAllester, D.: On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. IEEE Trans. Intelligent Transportation Systems, 12(2), 748–758 (2012)

20. Ren, S., He, K., Girshick, R., and Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. Proc. Advances Neural Information Processing Systems, pages 91–99 (2015)
21. Sabokrou, M., Fayyaz, M., Fathy, M., and Klette, R.: Deep-cascade: Cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes. IEEE Trans. Image Processing, `ieeexplore.ieee.org/document/7858798/` (2017)
22. Uijlings, J.R.R., Van De Sande, K.E.A., Gevers, T., and Smeulders, A.W.M.: Selective search for object recognition. Int. J. Computer Vision, 104(2), 154–171 (2013)
23. Viola P.,and Jones M.: Robust real-time face detection. Int. J. Computer Vision, 57, 137–154 (2004).
24. Wu, Y.N., Si, Z., Gong, H., and Zhu, S.-C.: Learning active basis model for object detection and recognition. Int. J. Computer Vision, 90(2), 198–235 (2010)
25. Xu, Y., Yu, G., Wu, X., Wang, Y., and Ma, Y.: An enhanced Viola-Jones vehicle detection method from unmanned aerial vehicles imagery. IEEE Trans. Intelligent Transportation Systems, `ieeexplore.ieee.org/document/7726065/` (2016)
26. Zhang, Y., and et al.: Vehicles detection in complex urban traffic scenes using Gaussian mixture model with confidence measurement. IET Intelligent Transport Systems, 10(6), 445-452 (2016)