IET Image Processing

Research Article

IET Journals
The Institution of Engineering and Technology

Turning video into traffic data – an application to urban intersection analysis using transfer learning

ISSN 1751-9659
Received on 4th August 2018
Revised 6th December 2018
Accepted on 6th January 2019
E-First on 6th March 2019
doi: 10.1049/iet-ipr.2018.5985
www.ietdl.org

Bhaskar Dey[1] ✉, *Malay Kumar Kundu*[2]

[1]*Center for Soft Computing Research, Indian Statistical Institute, 203 B.T. Road, Kolkata, India*
[2]*Machine Intelligence Unit, Indian Statistical Unit, 203 B.T. Road, Kolkata, India*
✉ *E-mail: bhaskar.dey09@gmail.com*

Abstract: With modern socio-economic development, the number of vehicles in metropolitan cities is growing rapidly. Therefore, obtaining real-time traffic volume estimates has a very important significance in using the limited road space and traffic infrastructure. In this study, the authors present a video-based traffic volume and direction estimation at road intersections. To discriminate the vehicles from the remaining foreground objects, vehicle recognition is performed by training a deep-learning architecture from a pre-trained model. This method, called transfer learning, primarily circumvents the requirement of huge labelled datasets and the time for training the network. The video sequence is first detected for moving foreground regions or patches. The trained model is subsequently used to classify the vehicles. The vehicles are tracked, and trajectory patterns are clustered using standard techniques. The number and direction of vehicles are noted, which are later compared with the manually observed values. All experiments were performed on real-life surveillance sequences recorded at four different traffic intersections in the city of Kolkata.

1 Introduction

Information about traffic volumes on various road segments and intersections is critical for transportation agencies, local administration, traffic census and historical trend analysis, and transport infrastructure planning/development. Measurement of traffic volume is indicative of congestion, possible air-pollutant concentrations, and projected fuel-tax and toll collections. A set of other traffic parameters such as vehicle-speed and vehicle-type classification has gained popularity as pavement design models are becoming increasingly sophisticated and new types of sensors are being introduced for highway operations. For many private/retail sector enterprises, obtaining accurate information at entries and exits for vehicles is essential. Authorities in high-traffic public places such as stadiums, airports, exhibition halls, entertainment centres/parks, shopping malls, museums, libraries, and bus/railway stations greatly benefit from an intelligent vehicle counting tool to implement better marketing strategy, optimise resource allocation etc.

There are many counting methods available. Broadly classifying, there are two methods, namely intrusive and non-intrusive. While the former involves a method that involves placing sensors on the roadbed, the latter is based on remote observation. Currently, the traditional intrusive technologies such as bending plate, pneumatic road tube, a piezo-electric sensor, and inductive loops are more widely applied in traffic data collection than non-intrusive technologies. This is in part due to their relative ease of use and institutional inertia. However, a new breed of non-intrusive technologies is being developed, which are not only reliable but easier to deploy/use. The advantage of non-intrusive sensors, when compared with their intrusive counterpart, is to minimise interference with traffic flow during sensor installation. Additionally, in bridges and tunnels where pavement cutting and boring are undesirable, non-intrusive sensors pose a viable alternative. Manual counts are the easiest option for non-intrusive traffic volume data collection. They are, however, labour intensive. Even if an enumerator monitoring traffic video has the sufficient dexterity and field of view, doing so for extended periods is extremely fatiguing. Other non-intrusive methods that are becoming more widespread include video traffic-detection systems, passive/active infrared detectors, microwave radar, ultrasonic detectors, and laser detectors. All these devices represent an emerging field of application that is expanding rapidly with advances in sensor technology and computer vision.

Vision-based techniques have been applied to vehicle-detection systems for over a decade. One of the earliest works in this field is the Autoscope® [1] vehicle-detection system that has been commercially deployed for traffic detections. In 2003, Veeraraghavan *et al.* [2] presented an overview of computer vision algorithms for intersection monitoring. There have been several recent investigations on vehicle classification using computer vision. Kamijo *et al.* [3] proposed a method for vision-based traffic monitoring and accident detection at intersections. In this method, the vehicle detection and tracking are modelled as a labelling problem for each pixel in the image sequence. To detect vehicles from static images, Wu *et al.* [4] used a pattern classifier based on the principal component analysis (PCA). Wavelet transform has been used to extract texture features for PCA. In addition, Sun *et al.* [5] applied gabor filters to extract textures features and subsequently verified each vehicle candidate using support vector machines. In [6], Lipton *et al.* used a classification metric to classify moving targets into vehicles, humans, and background clutter. Ohn-Bar and Trivedi [7] demonstrated the importance of learning appearance patterns of vehicle categories for identification and orientation estimation. A regression-analysis-based counting and classification of vehicles are presented in [8]. Zhuang *et al.* [9] used Haar-like features in combination with motion detection and a cascade of classifiers for faster vehicle detection. Recently, Leone *et al.* [10] proposed a cooperative visual sensor network for applications connected to the estimation of vehicular flows and parking management. In [11], counting, behaviour, and safety information extraction framework for traffic video was presented. Despite these great efforts, most vehicle-detection/counting applications are limited in a sense that they do not discriminate moving vehicles from other moving objects such as pedestrians, cycles etc., in congested traffic. This is because of the usage of a limited set of low-level hand-crafted features for vehicle detection. Of late, deep-learning architectures have shown promising results in many visual recognition tasks [12–14] including traffic-flow prediction [15]. Battiato *et al.* [16] presented a vehicle tracking method using a customised template matching. In [17], a thorough review of the application of computer vision techniques for traffic

analysis is given. Compared to methods which depend on hand-crafted features for object recognition, deep-learning methods do not require us to define and compute specific features for a given object. The lack of dependence on prior domain knowledge and effort in designing domain-specific features is a major advantage.

In this paper, we propose a method for automatically obtaining information on traffic-flow statistics as well as their directionality at the road intersections, i.e. intersection analysis from traffic video. Only a few attempts have been made to solve this problem. Our objective is to develop automated statistics of the total number of motor vehicles (excluding two wheelers and other vehicles that usually do not attract toll) moving in or out of the connecting road segments of a traffic intersection. For the current application, we restrict ourselves to (i) closed-circuit television (CCTV) footages captured under daylight conditions and (ii) do not discriminate among the various sub-categories of vehicles. To this end, a number of sample image patches of moving objects usually encountered in a congested traffic scene, i.e. pedestrians and vehicles are extracted manually and trained using a pre-trained network model MobileNet. Using a standard background segmentation algorithm, foreground patches corresponding to moving objects are obtained and classified with the newly trained model and tracked using the extended Kalman filter (EKF). The trajectories of the detected vehicles are then clustered using trajectory clustering.

The rest of this paper is organised as follows. In Section 2, the proposed method is described. Section 3 describes the experimental results. Concluding remarks are described in Section 4.

2 Proposed method

Fig. 1 presents an overview of the proposed method. The overall process consists of the training and testing phases. As mentioned earlier, traffic sequences from various road intersections are collected and split into two sub-sequences, which are used for training and testing separately. Hence, this section is split into two sections. In the first section, the method adopted for fine-tuning a pre-trained model for classification of images into vehicles and non-vehicles is discussed. This is followed by a description of background segmentation, tracking, and trajectory clustering.

2.1 Transfer learning: training using a pre-trained network model

As discussed earlier, we use a deeply layered architecture of convolutional neural network (CNN) for classification of a given image into two categories, namely vehicles and non-vehicles. However, when we train a network from scratch, we encounter the following two limitations: (i) huge data required – since the network has millions of parameters, to get an optimal set of parameters, we need to have a lot of data. (ii) The huge computing power required – even if we have a lot of data, training generally requires multiple iterations and it is very demanding on the processor.

Fortunately, we can leverage the existing models already trained on very large amounts of data for a different task to suit our current application. Many research groups share the models they have trained for large-scale image classification challenges. Such models have been trained on millions of images and for hundreds of hours on powerful graphics processing units (GPUs). For the current application, we use MobileNet (trained on ImageNet) as a starting point for our training process, instead of training our own model from scratch. Called *transfer learning*, this is a machine learning method where a model developed for a given task is reused as the starting point for a second task by fine-tuning only a subset of parameters of the network. Transfer learning is popular in the field of deep learning, given the enormous resource and labelled data required to train new models from scratch.

It may be noted that MobileNet is an architecture which is more suitable for mobile and embedded-based vision applications, where there is a lack of computing power. Hence, MobileNet model, as trained on the ImageNet dataset, was adopted for the current video-based application due to its faster processing requirements.

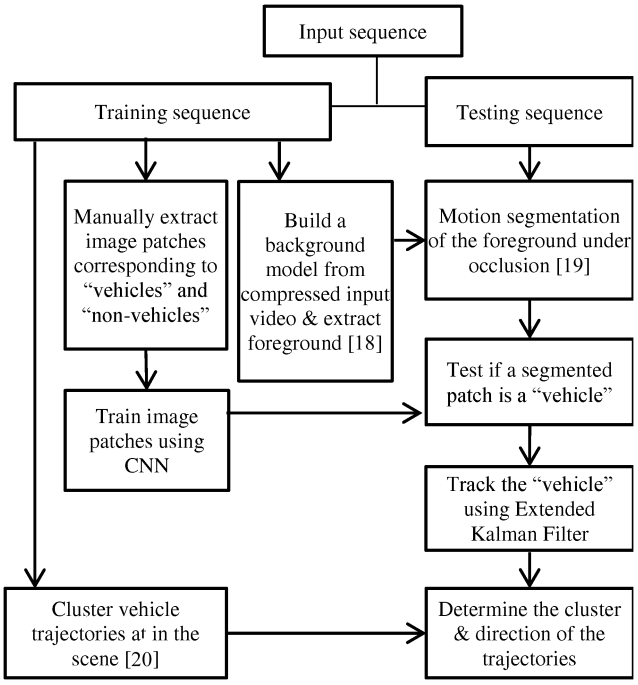

Fig. 1 *Schematic representation of the proposed method*

2.2 Vehicle classification using transfer learning

From the training sequences, image patches defined by the minimum-bounding rectangles corresponding to two classes of objects normally encountered in traffic, i.e. 'vehicle' and 'non-vehicle' are manually extracted. For each category, 3000 patches, which may be of different sizes, are obtained and rescaled to spatial dimensions of 224×224 for training with a CNN. We discuss the specific CNN architecture later. For the current application, motor-driven vehicles having three or more wheels are considered as members of the 'vehicle' class. Typically, these are the vehicles that attract toll and identification/counting of these vehicles is a prerequisite for proper video-based electronic toll collection and traffic regulation. Other moving objects such as pedestrians, bikes, cycles, manually driven vans etc. are considered as 'non-vehicles.' Fig. 2 shows illustrative examples from the two classes.

To extract and count the number of 'vehicles' in a traffic-intersection video, it is first required to segment the regions of interest (ROIs) corresponding to moving vehicles. As shown in Fig. 1, the proposed method requires foreground regions to be segmented from the background. Since the input videos are usually available in a compressed form, initially a fast background modelling method [18] is applied to the training sequence to obtain a background model. A key issue in a congested traffic scene is that the foreground segment of the frame obtained by background subtraction often remains occluded due to other objects present in the same scene. To count the number of vehicles correctly, it is required to segment independently moving objects as accurately as possible. It is a familiar experience, that when an object is occluded by another, both the objects appear adjacent to the imaging plane and share a common boundary/contour. Accurate segmentation in such cases, using a single image frame is difficult. Hence, the motion information of the occluding objects is obtained by measuring optical flow between two consecutive images of a sequence [19]. For example, Fig. 3 highlights the layer-wise motion segmentation of a given image containing objects under occlusion. The segmented patches corresponding to minimum-bounding rectangles of the object contours thus obtained are the candidates of either the 'vehicle' or the 'non-vehicle' class. To be able to classify such a patch during testing, we train a set of manually extracted patches with a pre-trained model of MobileNet.

As discussed earlier, the MobileNet model which was pre-trained on the ImageNet dataset on input images of size 224×224 was adopted. We used the pre-trained model for fine-tuning the network parameters suited to our application. The fine-tuning

a *b*

Fig. 2 *Example of patches extracted manually for training with CNNs*

(a) Some examples of manually extracted image patches which correspond to the vehicle class, ***(b)*** Some examples of manually extracted image patches which correspond to the non-vehicle class. These are objects encountered in a traffic scene but are not considered a vehicle for the current application, namely pedestrians, two wheelers (bicycles/motorbikes), and three wheelers that are not driven by a motor engine

a *b*

Fig. 3 *Annotation of occluded objects in motion using optical flow [19]*

(a) Image with moving objects under occlusion, ***(b)*** Optical flow-based segmentation of the image into ROIs. ROIs are coded in different colours to show the similarity/dissimilarity in motion

approach avoids both the limitations discussed above. The amount of data required for training is not much because of two reasons. First, we are not training the entire network. Second, the part that is being trained is not trained from scratch. Since the parameters that need to be updated is less, the amount of time needed will also be less.

For a given network architecture, the initial layers learn very general features and as we go higher up the network, the layers tend to learn patterns more specific to the task it is being trained on. Thus, for fine-tuning, we keep the initial layers intact (or freeze them) and retrain the later layers for our two-class problem. Our modified architecture is given in Table 1.

We used the MobileNet model as our base network. We freeze the entire network except for the last three layers, which are two-dimensional convolutional, batch normalisation, and an activation layer in sequence. In other words, only the last three layers of the existing network were marked trainable for training. The output of the network is flattened to a vector. It is appended to a fully connected layer with rectified linear unit (ReLU) [21] activation function. This is followed by a dropout (=0.5) layer [22, 23] and a softmax layer with two outputs. It may be noted that a large portion of the modified network has not been trained. Therefore, the number of parameters to train is thus greatly reduced.

Our training dataset is typically much smaller when compared with ImageNet; hence, it is compensated by setting a high dropout rate of 50%. The final layer has two output units corresponding to two output classes (i.e. vehicle and non-vehicle). The output class probabilities are predicted based on the softmax activation function.

Table 1 Modified network architecture used

Layer index	Layer name (type)	Output shape	Parameter number
0	mobilenet_1.00_224 (model)	(none, 7, 7, 1024)	3,228,864
1	flatten_1 (flatten)	(none, 50,176)	0
2	dense_1 (dense)	(none, 1024)	51,381,248
3	dropout_1 (dropout)	(none, 1024)	0
4	dense_2 (dense)	(none, 2)	2050

2.3 Processing of trajectories

It may be noted in Fig. 1, the ROIs identified as containing objects of the 'vehicle' class are tracked using the EKF. The centroid of the minimum-bounding rectangle corresponding to each ROI is used as a state variable for tracking. Furthermore, for each vehicle, a constant velocity is assumed. For a given sequence, each of the road segments connected to the intersection is first identified and named with letters. The centroid of the 'vehicles' forms trajectories which are clustered using [20]. To determine the number of vehicles entering/leaving the road segments of a traffic intersection and for subsequent evaluation, we demarcate and name each of them manually for identification (shown in Fig. 4) as point locations, A, B, …, and so on, on the image plane. During the testing phase, the number of 'vehicles' belonging to a particular cluster and the direction is recorded. For a given vehicle trajectory, the direction of the vehicle is inferred based on the minimal Euclidean (or straight line) distance between the entry/exit locations of a given vehicle trajectory and the demarcated point locations. Let a trajectory T_j be represented as $T_j = \{(x_j^i, y_j^i); i = 1, \ldots, N_j\}$, where (x_j^i, y_j^i) is the estimated position of the jth target on the image plane, N_j is the number of trajectory points, and $j = 1, \ldots, J$. J is the number of trajectories. Furthermore, let (A_x^k, A_y^k), $k = 1, 2, \ldots, N$ be the coordinates of the demarcated points on the image plane. For a given vehicle trajectory T_j having the starting location at (x_j^1, y_j^1) and the ending location at $(x_j^{N_j}, y_j^{N_j})$, the start and the end locations are assigned to two of the N demarcated locations as

$$\arg \min_{k = 1, 2, \ldots, N} \left(\left(x_j^1 - A_x^k\right)^2 + \left(y_j^1 - A_y^k\right)^2 \right) \quad (1)$$

and

$$\arg \min_{k = 1, 2, \ldots, N} \left(\left(x_j^{N_j} - A_x^k \right)^2 + \left(y_j^{N_j} - A_y^k \right)^2 \right) \quad (2)$$

respectively. The above formulation is based on the premise that the start and end locations of most trajectories are closest to the demarcated points in the image. Fig. 5 illustrates a tracked vehicle, its trajectory (shown with a dotted line from entry to exit locations), and the demarcated points A, B, C, and D for the 4-way intersection.

3 Experiments and discussion

3.1 Data description

For the current application, it was found that the number of existing datasets depicting intersection scenes along with ground-truth volume estimates for each road segment is severely limited. Moreover, a sufficiently large number of training images for the non-vehicle class is not publicly available. Therefore, for our specific application, we obtained a real-life dataset from the traffic-control authority managing busy traffic in the city of Kolkata, India. Our dataset consists of four video sequences, each of duration 15 min, captured at 24 fps from far-field traffic intersections with fixed charge-coupled device cameras. The video sequences were captured under daylight conditions at four different traffic intersections in Kolkata, namely Chandni Chowk, Rajabazar, Kankurgachi, and Shyambazar. The video sequences were obtained in the H.264 encoding format. There is a myriad of activities and interactions in the video data involving both pedestrian and vehicle movements. Please note that the current application uses vehicle detection for traffic-intersection analysis at congested places. Its primary objective is to distinguish motor-driven vehicles from pedestrians and manually driven vehicles in the presence of occlusion and count them while moving along the road segments of a traffic intersection. The proposed method is a specialised application unlike [7, 8], which are proposed for sub-categorisation of vehicle types and not for computation of vehicle flow along road segments. Thus, a comparison of these methods for vehicle detection would be unfair and hence not addressed here. For an intermediate-level comparison with reference to the identification of vehicles from non-vehicles, we compared our vehicle classification performance with those of [9].

3.2 Training data and augmentation

As mentioned earlier, each of the four sequences was divided into two parts: the duration of the first part is 5 min, which is used for training; the remaining part of the sequence is used for testing, i.e. traffic volume measurements at the intersections for the next 10 min. The training data, which consists of 2033 and 2177 images, respectively, from the vehicle and the non-vehicle categories, was obtained manually by cropping minimally bounded images for individual objects. The images of the vehicles and the non-vehicles classes were selected such that (i) the objects are not occluded and (ii) the overall training set is balanced in terms of the number of images having a particular type and positional view of the objects. The cropped images were resized to $224 \times 224 \times 3$ for training. This is followed by the process of data augmentation, which is conventionally used with deeply layered architectures in order to reduce overfitting. Data augmentation was performed by replicating each input image with horizontal (lateral) inversion. It may be noted that this process increases the number of training samples by a factor of two.

3.3 Experimental evaluations

The evaluation is done in two stages. First, the trained model is evaluated against a set of 2006 testing images obtained from the testing part of the sequences. The training module was implemented using Python, while the MobileNet model was imported from the Keras package. In the modified CNN architecture, the ReLU activation function has been used for all the layers. We trained our network for 40 epochs with batch sizes of 50. The training took roughly 3 h in total on a 3.50 GHz processor. Images from the vehicle and the non-vehicle categories were used for testing. For faster processing, the input frames were resized to 640×360. The overall execution time during testing, without using the GPU, was found to be 26 fps on the average. The classification result for the test images is illustrated in Table 2 (left) as a confusion matrix. The results were also compared with the implementation of [9]. Accordingly, AdaBoost classifier was trained on the same set of images as used for the proposed method. We collected 2033 positive samples (vehicle) and 2177 negative samples (non-vehicle) and uniformly resized the samples to a specific size of 224×224 px^2. As suggested in [9], we applied five-fold cross-validation on the dataset. The training with 3500 iterations took 139 min to complete on core *i*7 processor with 8 GB GPU.

a *b* *c* *d*

Fig. 4 *Demarcation of the road segments at traffic intersections. The point locations are named as A, B, and so on, marked in white on the imaging plane*
(a) Chandni Chowk Crossing, Kolkata (4-way intersection), ***(b)*** Rajabazar Crossing, Kolkata (4-way intersection), ***(c)*** Kankurgachi Crossing, Kolkata (3-way intersection), ***(d)*** Shyambazar Crossing, Kolkata (5-way intersection)

a *b*

Fig. 5 *Tracked vehicle with the corresponding trajectory*
(a) Image of a moving object shown to be tracked with a minimum-bounding rectangle. Location: Rajabazar Crossing, Kolkata, ***(b)*** Background of the scene shown with entry/exit points of the intersection. Trajectory of the tracked car is shown with the dotted line

Table 2 Confusion matrices for classification of images into vehicles and non-vehicles

	Total patches = 2006	Actual		Accuracy, %
		1 (non-vehicle)	2 (vehicle)	
***(a)** Confusion matrix obtained with the proposed method*				
predicted	1 (non-vehicle)	983	20	97.4
	2 (vehicle)	32	971	
***(b)** Confusion matrix obtained with the method* [9]				
predicted	1 (non-vehicle)	857	146	87.4
	2 (vehicle)	106	897	

Accuracy of image classification is shown in the right-most column of each matrix. Results of the proposed method (left) are compared with an implementation of [9] (right)

Table 3 Traffic volume estimates for Chandni Chowk 4-point intersection, Kolkata

Direction (i)	Ground-truth volume (R_i)	Estimated result (r_i)
AA	0	0
AB	23	21
AC	70	67
AD	32	28
BA	25	20
BB	0	0
BC	28	22
BD	216	209
CA	67	61
CB	13	10
CC	0	0
CD	0	0
DA	2	0
DB	177	157
DC	4	2
DD	0	0
global	657	597

Table 4 Traffic volume estimates for Rajabazar 4-point intersection, Kolkata

Direction (i)	Ground-truth volume (R_i)	Estimated result (r_i)
AA	0	0
AB	1	0
AC	58	49
AD	6	4
BA	25	21
BB	0	0
BC	3	2
BD	111	97
CA	90	81
CB	0	0
CC	0	0
CD	6	4
DA	0	0
DB	94	88
DC	5	3
DD	1	0
global	400	349

Table 5 Traffic volume estimates for Kankurgachi 3-point intersection, Kolkata

Direction (i)	Ground-truth volume (R_i)	Estimated result (r_i)
AA	0	0
AB	0	0
AC	162	110
BA	123	89
BB	0	0
BC	53	47
CA	182	163
CB	0	0
CC	0	0
global	520	409

As shown in Table 2, each row represents the numbers of members of a predicted class (output class), while each column represents their counterparts from the actual class (target class). It is found the trained model could detect images of the 'vehicle' and the 'non-vehicle' classes with higher accuracy (=97.4%) when compared with [9].

In the secondary stage of our evaluation, the traffic volume estimates at each road segment of an intersection are evaluated. We used the testing part of the sequences corresponding to each of the four traffic intersections. The ground-truth data detailing the traffic volume statistics were noted manually. The ground-truth data for Chandni Chowk, Rajabazar, Kankurgachi, and Shyambazar have been compared with the estimated traffic volumes obtained by the proposed methods in Tables 3–6, respectively.

Fig. 6 shows the screenshots of segmented vehicles at each of the four intersections. In each figure, the location of tracked vehicles is detected and indicated as minimum-bounding rectangles. The labels against the bounding boxes indicate the order in which the vehicles were detected. To quantify the accuracy of the proposed method, the observed direction (i) of the vehicles, the actual traffic volume (R_i) or ground truth, and the estimated volume along that direction (r_i) are noted. For a given intersection, the accuracy of traffic volume estimates was measured as

$$\text{accuracy} = 1 - \left(\sum_i |R_i - r_i| / \sum_i R_i \right) \tag{3}$$

The accuracies of the proposed method based on the obtained results for Kankurgachi, Chandni Chowk, Rajabazar, and Shyambazar are 0.79, 0.91, 0.87, and 0.87 respectively. Thus, the effectiveness of the proposed architecture for vision-based intersection analysis is promising and manifested.

4 Conclusion

We proposed a deeply layered architecture for recognition of vehicles from other objects in a traffic-intersection video. The traffic volume estimates are used later to determine the traffic-flow statistics from a given intersection video. Even with a limited set of training images, the accuracy of vehicle detection of our method was 97.4% (Table 2) while that for traffic volume estimates at the intersections being 78%, in the worst case (Kankurgachi).

The current application, however, performs poorly in case of highly overlapped areas, where the view of one of the vehicles is obstructed by another vehicle. Another issue is to improve the detection results at night or under very low-illumination conditions. The overall performance and speed of computation, in such cases, could be improved by using a GPU-based multi-scale vehicle segmentation technique with an analysis of each vehicle for their direction, which could be interesting areas for our future work.

5 Acknowledgments

We gratefully acknowledge the support of the Deputy Commissioner of Police (Traffic), Kolkata, for a donation of the CCTV footages used in this research. Malay K. Kundu acknowledges the Indian National Academy of Engineering (INAE) for their support through INAE Distinguished Professor fellowship.

Table 6 Traffic volume estimates for Shyambazar 5-point intersection, Kolkata

Direction (i)	Ground-truth volume (R_i)	Estimated result (r_i)
AA	0	0
AB	1	0
AC	16	15
AD	8	5
AE	0	0
BA	0	0
BB	0	0
BC	8	6
BD	0	0
BE	0	0
CA	0	0
CB	0	0
CC	0	0
CD	0	0
CE	0	0
DA	43	39
DB	87	80
DC	0	0
DD	0	0
DE	0	0
EA	0	0
EB	3	1
EC	16	12
ED	0	0
EE	0	0
global	182	158

Fig. 6 *Screenshots of the segmented vehicles at traffic intersections. Vehicles are numbered in the order in which they were first detected in the scene*
(a) Chandni Chowk Crossing, Kolkata (4-way intersection), ***(b)*** Rajabazar Crossing, Kolkata (4-way intersection), ***(c)*** Kankurgachi Crossing, Kolkata (3-way intersection), ***(d)*** Shyambazar Crossing, Kolkata (5-way intersection)

IET Image Process., 2019, Vol. 13 Iss. 4, pp. 673-679

6 References

[1] Michalopoulos, P.G.: 'Vehicle detection video through image processing: the autoscope system', *IEEE Trans. Veh. Technol.*, 1991, **40**, (1), pp. 21–29

[2] Veeraraghavan, H., Masoud, O., Papanikolopoulos, N.P.: 'Computer vision algorithms for intersection monitoring', *IEEE Trans. Intell. Transp. Syst.*, 2003, **4**, (2), pp. 78–89

[3] Kamijo, S., Matsushita, Y., Ikeuchi, K.*, et al.*: 'Traffic monitoring and accident detection at intersection', *IEEE Trans. Intell. Transp. Syst.*, 2000, **1**, (2), pp. 108–118

[4] Wu, J., Zhang, X., Zhou, J.: 'Vehicle detection in static road images with PCA and wavelet-based classifier'. Proc. IEEE Intelligent Transportation Systems Conf., Oakland, CA, 25–29 August 2001, pp. 740–744

[5] Sun, Z., Bebis, G., Miller, R.: 'On-road vehicle detection using gabor filters and support vector machines'. IEEE Int. Conf. Digital Signal Processing, Santorini, Greece, July 2002

[6] Lipton, A.J., Fujiyoshi, F., Patil, R.S.: 'Moving target classification and tracking from real-time video'. Proc. IEEE Workshop on Applications of Computer Vision, Princeton, USA, 1998, pp. 8–14

[7] Ohn-Bar, E., Trivedi, M.M.: 'Learning to detect vehicles by clustering appearance patterns', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (5), pp. 2511–2521

[8] Liang, M., Huang, X., Chen, C.-H.*, et al.*: 'Counting and classification of highway vehicles by regression analysis', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (5), pp. 2878–2888

[9] Zhuang, X., Kang, W., Wu, Q.: 'Real-time vehicle detection with foreground-based cascade classifier', *IET Image Process.*, 2016, **10**, (4), pp. 289–296

[10] Leone, G.R., Moroni, D., Pieri, G.: 'An Intelligent Cooperative Visual Sensor Network for Urban Mobility', *Sensors*, 2017, **17**, (11), p. 2588

[11] Shirazi, M.S., Morris, B.: 'A typical video-based framework for counting, behavior and safety analysis at intersections'. Proc. IEEE Intelligent Vehicles Symp. (IV), Seoul, Korea, 2015, pp. 1264–1269

[12] Krizhevsky, A., Sutskever, I., Hinton, G.E.: 'ImageNet classification with deep convolutional neural networks'. Proc. Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, 2012, pp. 1097–1105

[13] Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for large-scale image recognition'. Proc. Int. Conf. Learning Representations, San Diego, USA, 2015

[14] Chatfield, K., Simonyan, K., Vedaldi, A.*, et al.*: 'Return of the devil in the details: delving deep into convolutional nets'. Proc. British Machine Vision Conf. (BMVC), Dundee, Scotland, 2011

[15] Lv, Y., Duan, Y., Kang, W.*, et al.*: 'Traffic flow prediction with big data: a deep learning approach', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (2), pp. 865–872

[16] Battiato, S., Farinella, G.M., Furnari, A.*, et al.* 'Vehicle tracking based on customized template matching'. 2014 Int. Conf. Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, 2014, pp. 755–760

[17] Buch, N., Velastin, S.A., Orwell, J.: 'A review of computer vision techniques for the analysis of urban traffic', *IEEE Trans. Intell. Transp. Syst.*, 2011, **12**, (3), pp. 920–939

[18] Dey, B., Kundu, M.K.: 'Robust background subtraction for network surveillance in H.264 streaming video', *IEEE Trans. Circuits Syst. Video Technol.*, 2013, **23**, (10), pp. 1695–1703

[19] Liu, C., Freeman, W.T., Adelson, E.H.*, et al.*: 'Human-assisted motion annotation'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Anchorage, USA, 2008, pp. 23–28

[20] Anjum, N., Cavallaro, A.: 'Multi-feature object trajectory clustering for video analysis', *IEEE Trans. Circuits Syst. Video Technol.*, 2008, **18**, (11), pp. 1555–1564

[21] LeCun, Y.A., Bottou, L., Orr, G.B.*, et al.*: 'Efficient BackProp', in Montavon, G., Orr, G.B., Müller, K.R. (Eds.): '*Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*' (Springer, Berlin, Heidelberg, 2012)

[22] Hinton, G.E., Srivastava, N., Krizhevsky, A.*, et al.*: 'Improving neural networks by preventing co-adaptation of feature detectors', arXiv:1207.0580, 2012

[23] Srivastava, N., Hinton, G., Krizhevsky, A.*, et al.*: 'Dropout: a simple way to prevent neural networks from overfitting', *J. Mach. Learn. Res.*, 2014, **15**, (6), pp. 1929–1958