Vehicle Type Detection by Convolutional Neural Networks

Miguel A. Molina-Cabello[(✉)], Rafael Marcos Luque-Baena,
Ezequiel López-Rubio, and Karl Thurnhofer-Hemsi

Department of Computer Languages and Computer Science,
University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
{miguelangel,rmluque,ezeqlr,karlkhader}@lcc.uma.es

Abstract. In this work a new vehicle type detection procedure for traffic surveillance videos is proposed. A Convolutional Neural Network is integrated into a vehicle tracking system in order to accomplish this task. Solutions for vehicle overlapping, differing vehicle sizes and poor spatial resolution are presented. The system is tested on well known benchmarks, and multiclass recognition performance results are reported. Our proposal is shown to attain good results over a wide range of difficult situations.

Keywords: Foreground detection · Background modeling · Convolutional neural networks · Probabilistic self-organizing maps · Background features

1 Introduction

Nowadays, research on video surveillance systems is considered a prolific area due to mainly the great amount of available data obtained from any corner of the world. Concretely, the automatic analysis of traffic scenes is particularly relevant since it is possible to detect and avoid traffic congestions, incident and some breaches of road worthiness requirements [11]. Thus, a high-level description of the road sequences which involves the position, speed and class of the vehicles is sufficient to provide useful information about road traffic [9].

Foreground detection is the first step in any generic traffic video surveillance system. There are many algorithms which can model the background. For example, the background of a general scene can be modeled by using a single Gaussian distribution [15] or with a self-organizing neural network [8]. Furthermore, if the scenario is well-known, different techniques can be applied in order to improve the performance. For example, in this particular case of a traffic sequence, there are techniques which consider several facets like foggy or snow conditions [13].

Once an object is detected and tracked along the sequence, a simple labeling task, which identify the type of the object in motion, could be carried out. However, this process is not as straightforward as it seems to be, because a

feature extraction task is required to identify uniquely each class. This module should extract as many significant characteristics of the objects as possible, which are the inputs of the label detector or classifier, which the aim of improving the predicted class. Texture or brightness variations are features that can be employed for this purpose [14]. Recently, the use of deep learning networks has managed to alleviate the feature detection problem, performing this process in an intrinsic way in the first layers of the neural network [4].

Thus, in this work a new proposal for vehicle type detection in traffic videos is presented. Our proposed system is an enhancement of our previous vehicle tracking system [7]. A new module is added to this system in order to detect the type of the vehicles which appear in the scene. To this end a special kind of Convolutional Neural Network (CNN) has been used, namely AlexNet [5]. It has been previously used in traffic monitoring tasks [1,16]. The standard procedure to employ CNNs (AlexNet in particular) to object recognition, provides the neural network with an input image where the object of interest occupies most of the image area. If there are occlusions, they must be caused by objects that can not be confused with the objects to be recognized. For example, it is allowed that trees partially block the sight of a vehicle [1]. In current applications of CNNs to vehicle detection, either the camera footage is manually segmented to ensure that these conditions are fulfilled [1], or a special sensor arrangement is implemented so that the acquired images always satisfy them [16].

However, in many practical situations the preprocessing of the video footage prior to processing by the CNN is a challenge of paramount importance. In this work we focus on this key issue for the specific application of vehicle type detection in traffic scenes. This kind of video sequences often exhibit vehicles which partially occlude other vehicles. Hence a segmentation procedure must be put in place in order to separate the foreground regions from the background, and split those regions into vehicles. After that, rectangular windows must be determined which contain only one vehicle, and an appropriate resizing has to be carried out to honor the image size requirements of the CNN (in the case of the AlexNet, 227×227 pixel RGB images). The situation is substantially worsened by two factors. First of all, the size of the vehicles varies largely from motorbikes to trucks, which means that it is difficult to find an appropriate scaling of the original video data in order to fit the CNN size requirement while maintaining a high recognition rate. Secondly, traffic videos typically have a poor resolution, so that the segmented vehicles are given by a rather small number of pixels. Obviously, this hampers the recognition ability of the CNN. Here we study these issues and propose new solutions for them.

The rest of the paper is organized as follows: Sect. 2 presents the system architecture where this approach is integrated and Sect. 3 sets out the classification framework describing how the convolutional neural network is applied. Section 4 shows the experimental results over a public traffic surveillance sequence and Sect. 5 concludes the article.

2   System Architecture

The proposed vehicle tracking system (Fig. 1) can be divided in three different stages: an initial stage where the objects are detected from the sequence of frames, a second one where the objects are tracked, and finally, they are classified according to the different types of vehicles considered.
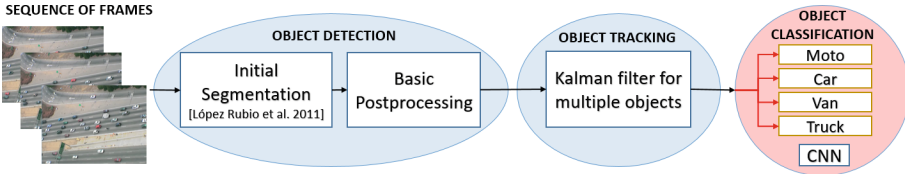

Fig. 1. Scheme of the proposed vehicle tracking system.

Initially, it is performed an initial segmentation based on the method proposed in [6], which it is based on the use of mixtures of uniform distributions and multivariate Gaussians with full covariance matrices, and it features an update process of the model that is based on stochastic approximation. One Gaussian component is used to model the background and one uniform component for the foreground, and when this probabilistic model is applied the Robbins–Monro stochastic approximation algorithm is used for the update equations [12]. It has been tested that it is a robust method for background modeling, and the stochastic framework has been proved to be suitable and effective as a learning method for real time algorithms with discarding data.

Then it is carried on a postprocessing applying some basic operators such as erosion and dilation. The first one carries out a superficial elimination of the borders of the objects. Thus, we can remove the false positives that are present in the segmented image, and also the false negatives which can appear inside the objects. Then, a dilation is applied to recover the original size of the objects. In many cases, objects that are very close to each other are represented with a single blob after the motion detection. These overlapping regions undergo a process that divides them to separate each individual vehicle. The final aim is to improve the quality of the segmented images and remove spurious objects.

In the tracking stage, a version of the Kalman filter for multiple objects has been implemented [10]. This statistical model computes the next position of the object using only information of the previous frame. In order to find the previous objects tracked, first the features of the objects detected in the new frame are extracted, and then a search over a window narrow centered in the centroids is performed. This Kalman model has as input the output obtained from the previous segmentation step.

At the end, the vehicles present in the image are classified depending on their features in four classes: moto, car, van and truck. This classification has carried out using the CNN as it is proposed in the following section.
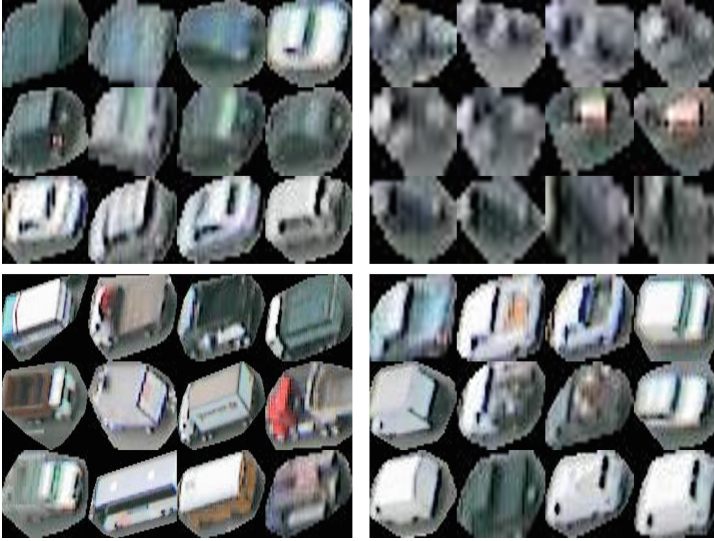
Fig. 2. Types of vehicles. A resize operation is applied in order to display them conveniently. First row shows cars and motorcycles, and second row reports trucks and vans, respectively.

3   Classification Framework

The CNN that we have employed in this work is AlexNet [5]. This net was developed in order to classify the IMAGENET dataset[1], which is a set of high-resolution images. Alexnet is composed by 5 convolutional layers and 3 fully-connected layers, where neurons which belong to a fully-connected layer have connections with all input neurons. In addition, normalization layers are used after the two first convolutional layers and its aim is that the excited neurons moderate their neighbor neurons. Finally, after these normalization layers and the fifth convolutional layer, the pooling layers have been placed. A Caffe replication of the original model is used in this work[2]. We have considered a set of input samples to train the network and this set presents the same number of samples for each vehicle class. On the other hand, the images have been selected from the testing sequence and each one of them show a motion vehicle from the video. Several vehicles from this dataset are shown in Fig. 2 ordered by its type.

In order to train the CNN, a dataset with all the images with the same size (256×256 pixels) is required. However, the vehicle samples does not have the same dimensions (e.g. trucks are bigger than motorcycles) which implies that all the image regions should be resized to serve as input to the neural network. So that, we need to apply a process to adapt the dataset to the needed size.

---

[1] http://www.image-net.org.
[2] https://github.com/BVLC/caffe/tree/master/models/bvlc_alexnet.

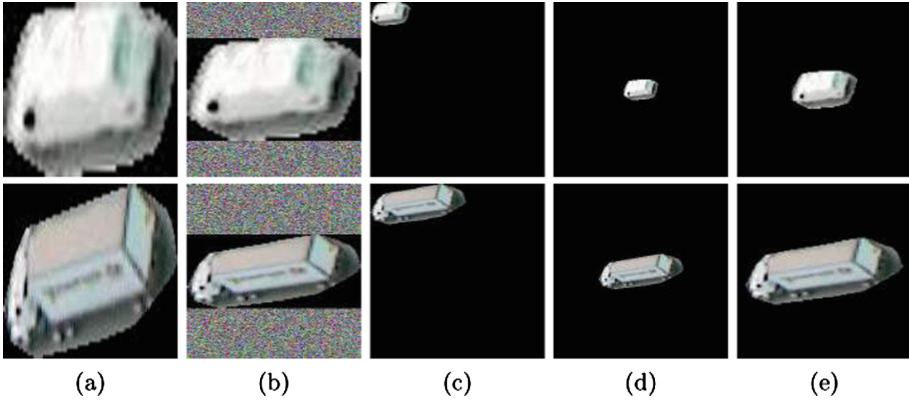(a)   (b)   (c)   (d)   (e)

Fig. 3. Resizing region approaches for 256×256 pixels. (a) Standard resize. (b) Resize preserving the aspect ratio. (c) Non-resize and non-centered. (d) Non-resize and centered. (e) Resize with scale and centered. First row presents a van while second row shows a truck. It is possible to observe that the two first strategies lose the size relation between the two vehicles, while the rest of approaches keep this visual proportion.

In this point, we have considered some different kinds of strategies in order to get the best possible performance. The first one is that we have noted the standard resize (*Standard*), where all the images have been resized to the required size, independently of its ratio. The second considered process (noted as *Fill*) applies an resize over the columns and it centers the vehicle, filling the rest of the pixels with noise. The third proposal (*Non-Centered*) considers the original size and fill the rest of the pixels with black color, placing the vehicle on the top left corner. The fourth approach (*Centered*) is like the previous one, but centered the vehicle on the image. The last suggested process (*Centered Scale*) resizes the images applying a scale preserving the ratio between the original images, placing it in the center of the image and the rest of the pixels are filled with black color. All of this proposals are reported in Fig. 3.

Deep learning networks work best if the number of input patterns is high. In cases where this circumstance is not fulfilled, it is possible to carry out a process called data augmentation. In this work we will take 10 regions of 227×227 pixels of each sample that will serve as input to the network, where each region will have a random displacement over the original region of 256×256 pixels. In addition to modifying the input of the network, it is necessary to modify the output. The only variation in this point is to change the number of output classes to 4, because there will be four types of vehicles to detect: cars, motorcycles, trucks and vans.

4   Experimental Results

In this section we report the different tests that we have carried out and its results. In order to test the proposal we show in this work, we have chosen a

Table 1. Quantitative measures of the results. Each row is a measure and each column presents the mean and the standard deviation. Best results are highlighted in **bold**.

Measure	Standard	Fill	Non-Centered	Centered	Centered Scale
Accuracy	0.684 ± 0.02	0.670 ± 0.01	0.772 ± 0.02	0.862 ± 0.01	$\mathbf{0.865 \pm 0.01}$
Mean Square Error	1.471 ± 0.11	1.387 ± 0.09	1.117 ± 0.07	0.913 ± 0.08	$\mathbf{0.893 \pm 0.11}$
Rand Index	0.764 ± 0.01	0.759 ± 0.01	0.822 ± 0.02	0.876 ± 0.01	$\mathbf{0.879 \pm 0.01}$
Hubert's Gamma	0.336 ± 0.03	0.327 ± 0.02	0.503 ± 0.04	0.650 ± 0.01	$\mathbf{0.651 \pm 0.03}$
Overall Cluster Entropy	0.622 ± 0.03	0.634 ± 0.01	0.461 ± 0.04	0.293 ± 0.01	$\mathbf{0.292 \pm 0.03}$
Overall Class Entropy	0.559 ± 0.03	0.560 ± 0.02	0.405 ± 0.04	$\mathbf{0.251 \pm 0.00}$	0.265 ± 0.02
Overall Entropy	0.590 ± 0.03	0.597 ± 0.01	0.433 ± 0.04	$\mathbf{0.272 \pm 0.01}$	0.279 ± 0.02

sequence which shows a highway with different kinds of vehicles moving on it and it presents some troubles like occlusions or overlapping objects. The selected video is named U*S-101 Highway*. This sequence is taken from the dataset of Next Generation Simulation (NGSIM) program, provided by the Federal Highway Administration (FHWA).

Our proposal tries to classify the detected vehicles of the sequence into 4 different classes depending on the characteristics of the vehicle. The possible classes are motorcycle, car, van and truck. In order to train the network and test it, we have selected several vehicles with its trajectory and we have labeled them. The trajectory of a segmented object O_i is a set of features $\{\boldsymbol{x}_i^f \in \mathbb{R}^4 \mid f \in [1..MaxFrame]\}$ that it is composed by all the frames where this vehicle appears in the whole video. Then, we form a set of images that we use to train and test the performance of our proposal. The images of this set corresponds to images of each selected trajectories previously.

In order to obtain a robust performance of the goodness of the proposal we have employed a 10-fold strategy, where the training set has the 90% of the data and the test set has the 10 remaining percent. Furthermore, each set has the same number of images of each class. In this way, we have repeated the process 10 times. The next step is the training of the network applying the different resizing region approaches that we have described in the section before.

After the training, we use the test set to measure the performance of the approach. In order to compare the proposals among themselves from a quantitative point of view, we have selected some different well-known measures. The most important considered measures in this work are the Accuracy (Acc), which attains values in the interval $[0, 1]$, where higher is better, and it represents the percentage of hits of the system; and the Mean Square Error (MSE), which is a positive real number, where lower is better, and it calculates the error of the model when it predicts a class for an object.

Table 2. Accuracy values for each labeled class (higher is better). Each row represents a different type of vehicle (car, motorcycle, truck and van). The studied resizing region proposals are shown from the second to the fifth columns. Last column displays the global accuracy for all the classes.

Class	Standard	Fill	Non-Centered	Centered	Centered Scale
Car	0.671 ± 0.02	0.683 ± 0.03	0.804 ± 0.03	0.803 ± 0.01	$\mathbf{0.829 \pm 0.03}$
Moto	0.883 ± 0.01	0.820 ± 0.05	0.811 ± 0.03	$\mathbf{0.994 \pm 0.01}$	0.992 ± 0.01
Truck	0.546 ± 0.03	0.578 ± 0.02	$\mathbf{0.918 \pm 0.02}$	0.893 ± 0.02	0.909 ± 0.01
Van	0.653 ± 0.05	0.614 ± 0.04	0.595 ± 0.03	$\mathbf{0.808 \pm 0.03}$	0.777 ± 0.04
All	0.684 ± 0.02	0.670 ± 0.01	0.772 ± 0.02	0.862 ± 0.01	$\mathbf{0.865 \pm 0.01}$

Let K be the existing objects and k the observed object, we note as $\mathbf{x}_k$ and $\mathbf{w}_k$ the real and the predicted class of the object k, respectively, where $\mathbf{x_k}, \mathbf{w_k} \in \{1, 2, 3, 4\}$, corresponding $1 = car, 2 = moto, 3 = truck$ and $4 = van$. Moreover, if the model succeeds the classification of the object k, (that is, $\mathbf{x_k} = \mathbf{w_k}$), we note this as $\mathbf{q}_k = 1$; and we note as $\mathbf{q}_k = 0$ when the model makes a mistake in its prediction (that is, $\mathbf{x_k} \neq \mathbf{w_k}$). So that, the definition of the Accuracy and the Mean Square Error are:

$$Acc = \frac{1}{K} \sum_{k=1}^{K} \mathbf{q}_k \qquad MSE = \frac{1}{K} \sum_{k=1}^{K} (\mathbf{x}_k - \mathbf{w}_k)^2 \qquad (1)$$

We have selected other measures which can be employed in order to complete the comparison of the classification performance of the proposed models. One of them is the Rand Index [3], which is a value between 0 and 1, and higher value is better. We have also considered the Hubert's Gamma Statistic [3], whose values are in the interval $[-1, 1]$ and higher values are better. Finally, we have included some entropy measures: the Overall Cluster Entropy, the Overall Class Entropy and the Overall Entropy [2]. This measures are values between 0 and 1, where lower is better.

The performance of each proposal according to these measures is reported in Table 1. As it can be observed, the Centered Scale process achieves the best performance in five of the seven considered measures. The Centered process also obtains a good performance, with the best result in the other two measures and similar values in the rest of the measures. These results reflect that the sizing strategies that preserve the difference in size between classes, are able to improve the accuracy values of the classification process. Therefore, a good performance is achieved despite the fact that the small number of labeled data and its low quality and size.

In addition, a comparative between each approach and its performance with each vehicle class is shown in Table 2. In this case, the Centered process gets the best performance in two of the four considered vehicle types. Nevertheless, the Centered Scale obtains the best mean from all the classes. As may be evident,
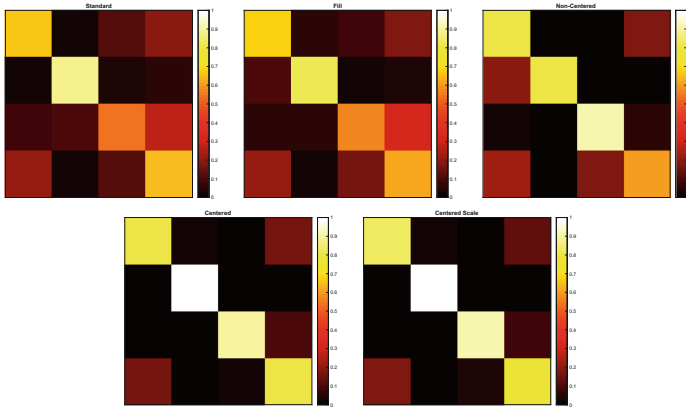
Fig. 4. Confusion matrices for the resizing region approaches, namely: Standard, Fill and Non-Centered (first row); Centered and Centered Scale (second row). The order of the classes for both columns and rows is: car, motorcycle, truck and van. The row dimension indicates the real class while the column dimension reflects the predicted class. The clearer the blocks on the diagonal the greater the accuracy of the method.

the greater complexity lies in the distinction between the Car and Van classes (80% of success on average), because of its undeniable similarity. In fact, visually there may be doubts in its correct classification, mainly due to the low resolution of the regions (see Fig. 3).

The confusion matrices for each resizing strategy, which show the errors between the predicted class by the model and the real class of a vehicle, are shown in Fig. 4. The clearer the blocks on the diagonal the greater the accuracy of the method. If a block outside the diagonal is not very dark, it implies that, given an image of a vehicle, there is an error between the output of the network (prediction) and the actual class to which it belongs. For example, it is noted that the blocks relating to the classes Car and Van display more orange tones than blocks relating to the classes Car and Motorcycle. In this way, both the Centered and the Centered Scale approaches provide visually the best result.

Additionally, another important result is that the distinction between vans and trucks cause certain difficulty. is very difficult. This is because there are some trucks which have a similar size with respect to a van. Moreover, the segmentation and the tracking processes cause negative effects in the reported results. This two steps introduce mistakes and occasionally provide a wrong segmentation, so that it produces a bad classification.

Finally, the best trained classification model has been integrated in the developed video surveillance traffic system previously described. Some qualitative results are reported in Fig. 5, where the online vehicle classification can be observed in several tested frames.
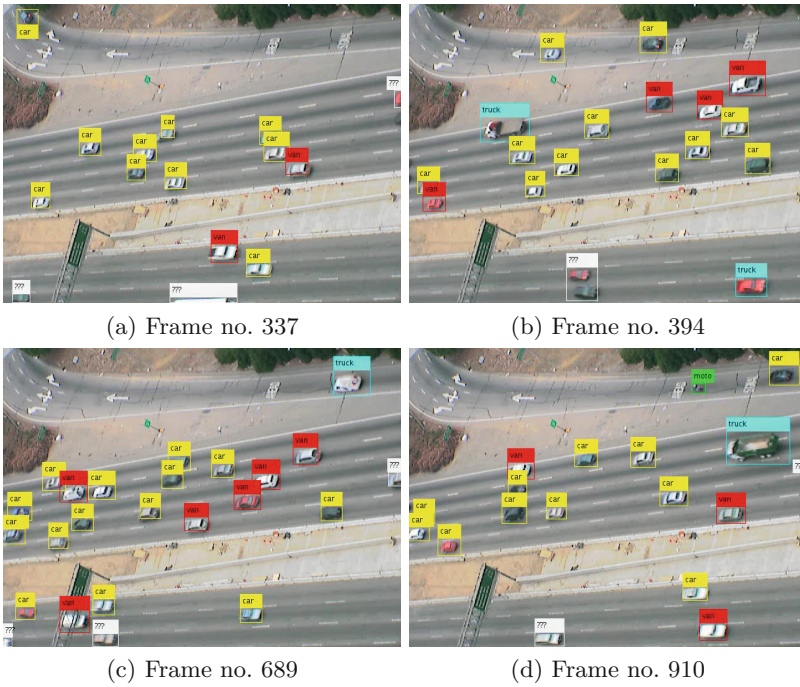
(a) Frame no. 337

(b) Frame no. 394

(c) Frame no. 689

(d) Frame no. 910

Fig. 5. Online vehicle classification of the sequence *S-101 Highway*.

5   Conclusions

We have proposed an approach employing a Convolutional Neural Network (CNN) AlexNet network model to classify the vehicles that are presented in traffic sequences. The system assigns each detected vehicles to a class: car, motorcycle, truck or van. Because the AlexNet network needs a training image dataset where all the images have the same size, different resizing region approaches have been considered to transform the dataset to adapt the size of its images to the required size. A quantitative comparison between these proposals have been studied. The reported results show the noted Centered Scale method as the best classifier with a high accuracy. In addition, the low resolution of the training dataset and the considered difference between classes car and van affect negatively to the obtained performance. Finally, we have integrated the best classification proposal to our previous developed vehicle tracking system with an online strategy in order to complete a real-time classification of the vehicles.

References

1. Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., Vairo, C.: Deep learning for decentralized parking lot occupancy detection. Expert Syst. Appl. **72**, 327–334 (2017)
2. He, J., Tan, A.H., Tan, C.L., Sung, S.Y.: On quantitative evaluation of clustering systems. In: He, J., Tan, A.-H., Tan, C.-L., Sung, S.-Y. (eds.) Clustering and Information Retrieval, pp. 105–133. Springer, New York (2004)
3. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Inc., Upper Saddle River (1988)
4. Kato, N., Fadlullah, Z.M., Mao, B., Tang, F., Akashi, O., Inoue, T., Mizutani, K.: The deep learning vision for heterogeneous network traffic control: proposal, challenges, and future perspective. IEEE Wirel. Commun. (2016)
5. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25, pp. 1097–1105 (2012)
6. López-Rubio, E., Luque-Baena, R.M.: Stochastic approximation for background modelling. Comput. Vis. Image Underst. **115**(6), 735–749 (2011)
7. Luque-Baena, R.M., López-Rubio, E., Domínguez, E., Palomo, E.J., Jerez, J.M.: A self-organizing map to improve vehicle detection in flow monitoring systems. Soft. Comput. **19**(9), 2499–2509 (2015)
8. Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance applications. IEEE Trans. Image Process. **17**(7), 1168–1177 (2008)
9. Mithun, N., Howlader, T., Rahman, S.: Video-based tracking of vehicles using multiple time-spatial images. Expert Syst. Appl. **62**, 17–31 (2016)
10. Reid, D.: An algorithm for tracking multiple targets. IEEE Trans. Autom. Control **24**(6), 843–854 (1979)
11. Ren, J., Chen, Y., Xin, L., Shi, J., Li, B., Liu, Y.: Detecting and positioning of traffic incidents via video-based analysis of traffic states in a road segment. IET Intel. Transport Syst. **10**(6), 428–437 (2016)
12. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**(3), 400–407 (1951)
13. Sen-Ching, S.C., Kamath, C.: Robust techniques for background subtraction in urban traffic video. In: Electronic Imaging 2004, pp. 881–892. International Society for Optics and Photonics (2004)
14. Wang, K., Liu, Y., Gou, C., Wang, F.Y.: A multi-view learning approach to foreground detection for traffic surveillance applications. IEEE Trans. Veh. Technol. **65**(6), 4144–4158 (2016)

15. Wren, C., Azarbayejani, A., Darrell, T., Pentl, A.: Pfinder real-time tracking of the human body. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 780–785 (1997)
16. Wshah, S., Xu, B., Bulan, O., Kumar, J., Paul, P.: Deep learning architectures for domain adaptation in HOV/HOT lane enforcement. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–7 (2016)