IET Intelligent Transport Systems

Research Article

Journals
The Institution of
Engineering and Technology

Vehicle detection, counting and classification in various conditions

ISSN 1751-956X
Received on 20th July 2015
Revised on 26th December 2015
Accepted on 7th February 2016
doi: 10.1049/iet-its.2015.0157
www.ietdl.org

Shiva Kamkar, Reza Safabakhsh ✉

Department of Computer Engineering and Information Technology, Amirkabir University of Technology (Tehran Polytechnic), Hafez Ave, Tehran, Iran

✉ *E-mail: safa@aut.ac.ir*

Abstract: Intelligent transportation systems have received a lot of attention in the last decades. Vehicle detection is the key task in this area and vehicle counting and classification are two important applications. In this study, the authors proposed a vehicle detection method which selects vehicles using an active basis model and verifies them according to their reflection symmetry. Then, they count and classify them by extracting two features: vehicle length in the corresponding time-spatial image and the correlation computed from the grey-level co-occurrence matrix of the vehicle image within its bounding box. A random forest is trained to classify vehicles into three categories: small (e.g. car), medium (e.g. van) and large (e.g. bus and truck). The proposed method is evaluated using a dataset including seven video streams which contain common highway challenges such as different lighting conditions, various weather conditions, camera vibration and image blurring. Experimental results show the good performance of the proposed method and its efficiency for use in traffic monitoring systems during the day (in the presence of shadows), night and all seasons of the year.

1 Introduction

With the increasing number of roads and traffic all over the world, traffic monitoring and control using modern technologies has become a compelling requirement. This is why intelligent transportation systems (ITS) have attracted a lot of attention in the last decades. Among different tasks in a traffic monitoring system, vehicle detection is the key task [1]. After a vehicle is detected, other applications can be applied more easily. In the past, active sensors such as lasers, lidar, or millimetre-wave radars were used to detect vehicles [2]. These sensors usually produced limited information on traffic parameters. Today, passive sensors are more common. They specially benefit from videos or images prepared by traffic cameras installed over the roads or on the roadside. These methods have lower cost and higher performance. Different traffic parameters such as vehicle type, the number of cars, traffic density, the average speed and even traffic accident information can be extracted only using traffic videos or images in a short time [3].

As most monitoring systems are based on video, we also based our study on traffic videos. First, we proposed a vehicle detection method which uses the active basis model (ABM) to detect vehicle candidates. This model benefits the edge information to first make a common sketch of vehicle types and then detects vehicle candidates by template matching. Edge information can be used in different lighting and weather conditions and is less sensitive to other challenges such as blurring and the effects of camera vibration in video frames. This makes the proposed method applicable to traffic videos all day and night long in various weather conditions. We verified vehicle candidates using the shape symmetry and also introduced a mechanism not to eliminate vehicles which do not seem symmetrical because of lane changes or the angle of view. In the next step, a line is defined to be used for counting vehicles passing it. Finally, we extracted a length-based and a texture-based feature from the counted vehicles and applied a random forest (RF) to classify them. The vehicles are classified into one of these three categories: small (e.g. car), medium (e.g. van) and large (e.g. bus and truck). The counting and classification steps can be used in real-time applications due to their trivial run-time.

The paper is organised as follows. Section 2 provides a review on the latest related research. We describe the proposed vehicle detection, counting and classification methods in Section 3. In Section 4, the experiments are described and Section 5 concludes the paper.

2 Related work

The Jet Propulsion Laboratory used a computer vision method to detect vehicles for the first time in 1978 [3]. Since then, various studies have been carried out to improve the performance of different parts of ITS [4–7]. There are two main approaches in vehicle detection. The first one uses vehicle motion information and the second one uses the inherent features of vehicles to detect them in videos.

The basic idea behind the motion-based approach is that in a traffic scene, everything is static except for vehicles. So vehicle candidates can be identified by determining moving parts of the scene. Motion-based methods have some challenges which make them inappropriate in certain situations. As the basic idea is based on the motion of vehicles, they hardly detect static vehicles. Another challenge refers to the motion of other parts of the scene. It is common for a road to have moving things other than vehicles such as pedestrians, leaves of trees or birds. Even rain and snow may have similar effects. In addition, lighting changes during the day may result in moving shadows in the scene which come along with vehicles and may be determined as a part of them. Furthermore, at night, when a vehicle's headlights light up its front area, they actually create another moving region. Many researchers have proposed methods to overcome the above challenges. Some researchers have presented different background subtraction algorithms for improving moving object segmentation in a video [8, 9]. Others have proposed novel methods for background estimation [10, 11]. One researcher focused on heavy traffic and proposed an adaptive approach for vehicle detection in traffic jams and in complex weather [12]. Another researcher suggested an approach using a full-search block matching method that detects the moving part and can prevent false motion using adaptive thresholding in a challenging environment [13]. An improved version of the background subtraction method has also been suggested to alleviate the negative impacts from gradual changes. Then, a level set method has been used to determine blobs and finally a Kalman filter and support vector machine are utilised to improve the accuracy of vehicle classification [14].

The basic idea behind the inherent features-based approach is that vehicles have special features which can be used to differentiate them from other objects in traffic scenes. The methods use various features such as colour, texture, edge information and shape symmetry to find vehicle candidates. These methods can be applied to both traffic videos and images and do not have the challenges of the motion-based approach, but some are time consuming. Using the inherent features of vehicles, researchers also proposed a novel colour-based method to detect vehicles from images. They presented a statistic linear model of colour change space to compact vehicle colours and thus narrow down the search areas of possible vehicles [6]. Other researchers proposed a generative model based on edges and constructed a Markov chain Monte Carlo method to detect and segment vehicles in static images [15]. Some researchers presented night time vehicle detection algorithms for traffic surveillance by locating and analysing some features of vehicle lights [16, 17]. In another work, vehicle detection in videos is carried out by performing matching between a vehicle model based on the mixture of deformable part models to the features collected by sliding windows of a histogram of oriented gradients [18]. Recently, some researchers constructed a multi-scale model to detect vehicles with various distances from the camera in traffic images [19]. They also constructed a special and-or graph, hybrid image template and a part-based model to represent and detect vehicles in images with a congested traffic condition [20, 21]. In another research, a deformable model, the ABM, is constructed from a set of training images to detect vehicles in new traffic images by template matching. To improve the performance in vehicle recognition, the edge and colour symmetry features of the vehicle are used [22]. There are other studies which combined both inherent features like colour and edge and the motion information [23, 24].

Vehicle counting methods usually specify an area and check if any vehicle enters this area [9, 25, 26]. Those investigations including vehicle classification extract shape-based features like length, width and area [25–28] and then use a classifier such as the *k* nearest neighbour method or a neural network to categorise the vehicles.

3 Proposed method

The method proposed in this paper to detect, count and classify different types of vehicles is illustrated in Fig. 1.

For vehicle detection, the method constructs an ABM [29]. The model processes training images of different types of vehicles and constructs a general sketch of them. Then the sketch is used to find vehicle candidates in the video frames. We use the sketch matched on each candidate to verify it and check whether it is symmetrical or not. For vehicle counting, we use a single counting line. Vehicles passing this line are counted and are given to the classification section to be classified into one of the three classes. We extract the vehicle length in the time-spatial image (TSI) and also correlation of grey-level co-occurrence matrix (GLCM) obtained from the vehicle's bounding box. The extracted features are given to a RF classifier for categorisation. In the following sections, we explain each step in more details.

3.1 Vehicle detection

First we detect vehicle candidates using ABM. This model creates a common sketch from a small number of training images. Since we only focus on vehicles with more than three wheels, training images are captured from the top and front views. Different cars, vans and trucks or buses have more common structures in this view. When a vehicle is detected as a candidate by this algorithm, we extract it within its bounding box and check its reflection symmetry by calculating the correlation value. The ABM and symmetry evaluation algorithms are further explained below.

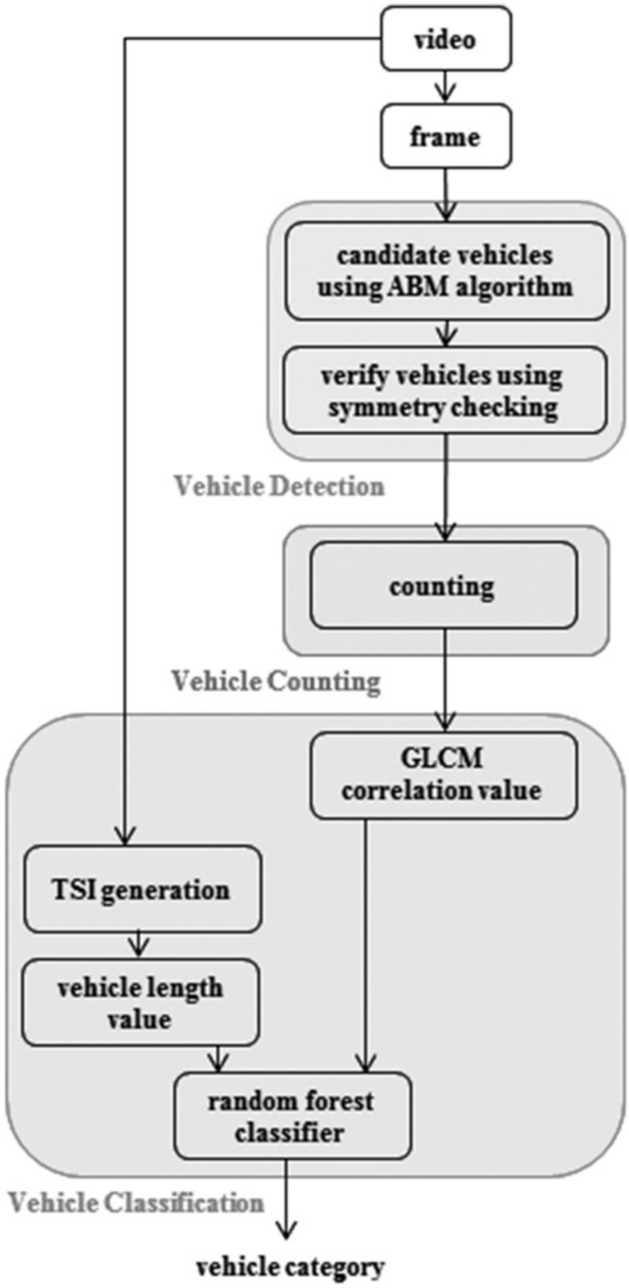


Fig. 1 *Proposed method flowchart*

3.1.1. Active basis model: ABM is a powerful object detection method which can create a general sketch of objects after processing a small number of training images. An active basis is made of a small number of Gabor wavelet elements at specific locations with specific orientations. The elements can include minor changes in locations and orientations. Active basis is formed by placing the elements by each other linearly. Fig. 2 illustrates a sample active basis. Gabor wavelet elements are shown by black ellipsoids. Each element can be converted into the grey ellipsoid by perturbing the location and orientation, so the active basis is deformable.

A sample learned ABM is shown in Fig. 3. The model is learned from nine training images of cars within their bounding boxes. The location and orientation of each element are learned from the corresponding element in the training active bases by the shared sketch algorithm proposed by Wu *et al.* [29]. As shown in Fig. 3,

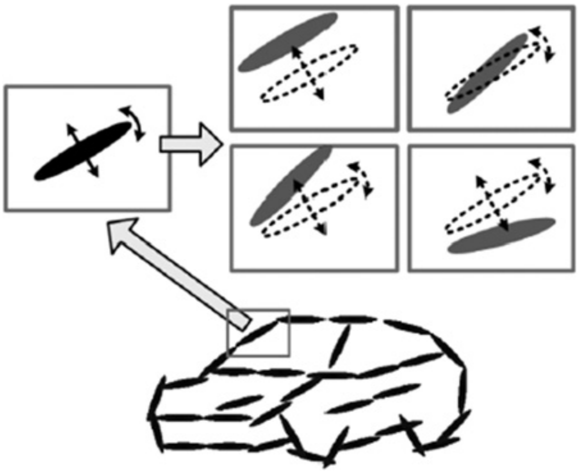


Fig. 2 *Sample active basis consists of Gabor wavelet elements [29]*

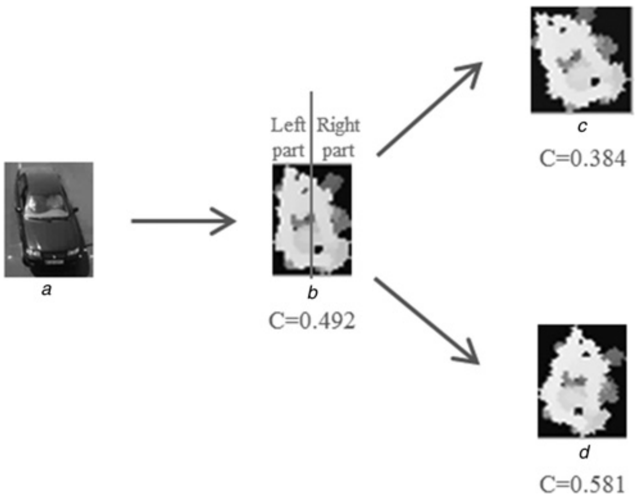


Fig. 4 *Rotation makes some diagonal vehicles seem more symmetrical*

a Detected vehicle
b Active basis of detected vehicle
c After counter clockwise rotation
d After clockwise rotation
Correlation (C) between left and right parts is computed for each image

the ABM has successfully represented a common sketch for vehicles according to the training data. The elements of the common sketch show little overlap and are well connected.

The method is not sensitive to various backgrounds of training images. In addition, as it is an edge-based algorithm, lighting changes, weather changes, small video quality changes, even camera movements and image blurring have ignorable effects on the detection performance. Therefore, it is usable in real and normal traffic situations.

After the ABM is formed, template matching is applied to find parts of the frame that are similar to the model. Active correlation is used to measure the similarity. We have considered a threshold value for the correlation between the common template and parts of the frame. The threshold prevents the algorithm from introducing less similar parts of the frame as a vehicle. As some frames of the video include no vehicle, defining this threshold is very effective in decreasing the false positive results. Vehicle detection is applied to videos in only one resolution on every five frames in order to reduce the computational load and run-time with inconsiderable effects on detection accuracy.

3.1.2 Symmetry evaluation: When an object is declared as a vehicle candidate by the above method, the common sketch deforms to match that particular vehicle candidate. Reflection symmetry is one of the properties of most man-made objects especially vehicles. We use this feature to verify vehicle candidates. After the extraction of the deformed common sketch, we divide it into two parts according to the vertical axes from the middle. Then we calculate the correlation between the left part and the mirror image of the right part. We accept candidates whose symmetry is higher than a threshold. However, we do not completely reject those candidates with a small correlation. Due to the large width of the highways and the lane changing of vehicles, there are some vehicles which are not completely symmetrical from the front view. Thus, when a vehicle does not hold the symmetry condition, we rotate it in the clockwise and counterclockwise directions and check the symmetry condition again (Figs. 4*a–d*). If the condition holds, we accept it as a vehicle; otherwise, we reject it.

Threshold and rotation angle are parameters which should be determined experimentally. After applying the above algorithms to each frame, all vehicles within their bounding boxes are ready to be counted. The counting procedure uses the results of the detection step. Thus, parameter determination and the accuracy of this step have direct effects on the counting procedure performance.

3.2 Vehicle counting

For counting vehicles, we used only one counting line. We define a line in about the middle three-fifth of the frame where vehicles are expected to be detected. For each vehicle coming towards the camera, two consequent frames exist where in the first frame the vehicle centre is before the line and in the second one; it is on or after the line. It is exactly at this frame that we count the vehicle. We defined a *Tracking List* containing the centres of vehicles bounding boxes which are before the counting line. The *Tracking list* items are updated until the vehicles pass the line. Then we count and remove them from the *Tracking List*. The counting algorithm is given in Fig. 5.

Values of *threshold* and n are calculated experimentally. The proposed method avoids mistakes in counting occluded vehicles,

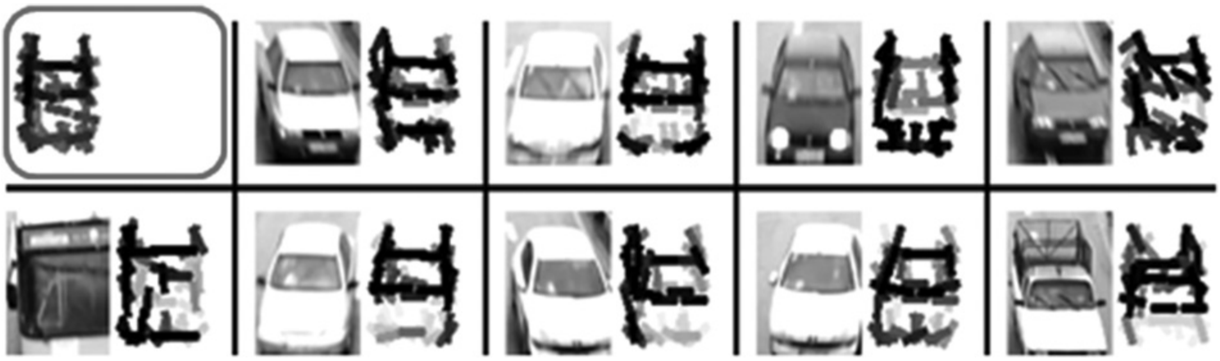


Fig. 3 *Active basis learned from nine training images. Learned active basis is given in the upper left rectangle*

```
For each frame f of video
  For each vehicle detected v in f
    Find vc(vx,xy) = center of vehicle bounding box
    If vx< DetectionLine
      For each member m(mx,my) from TrackingList
        If distance(vc,m) < threshold and distance(vx,mx) > 2* distance(vy,my)
          Replace vc with m in TrackingList
        Else
          Add vc to TrackingList
    Else
      For each member m(mx,my) from TrackingList
        If distance(vx,m) < threshold and distance(vx,mx) > 2* distance(vy,my)
          Count vc as a vehicle
          Remove m from TrackingList
  End
  Check TrackingList and remove members that have not updated after passing n
frames
End
```

Fig. 5 *Vehicle counting algorithm*

because the occluded vehicles are detected separately in the vehicle detection algorithm and their bounding boxes are saved separately in the *TrackingList*. We update a vehicle in the list in case that its motion direction is along the road, so occluded vehicles will be tracked and counted separately. In addition, we remove the vehicles bounding boxes which are not updated after passing n frame in order to avoid counting a vehicle more than once. The bounding box of a vehicle which is counted is given to the classification part for further processing.

3.3 Vehicle classification

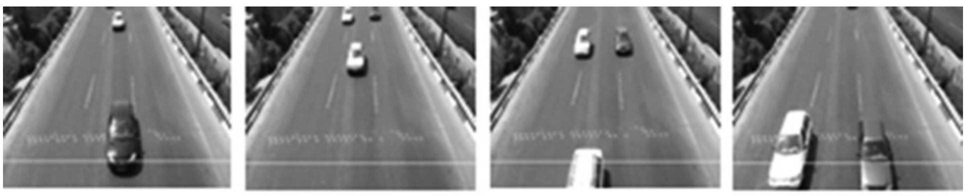
In the classification step, we classify vehicles into three classes: small (e.g. car), medium (e.g. van) and large (e.g. bus and trunk). To reach this goal, we extract two features to differentiate between different vehicle types. First, we compute a length-based feature that is very useful for classifying vehicles according to their size. However, this feature is sensitive to noise and the vehicle velocity changes. As a result, we add a texture-based feature to improve the classification accuracy. Then, a RF is used to classify vehicles. The details of these steps are described in the following sections.

3.3.1. Vehicle length in TSI: TSI of a video sequence is generated using a line on frames in which a vehicle passes the line. The line is perpendicular to the motion of the vehicles. We benefited from the counting line used in the vehicle counting procedure for this purpose. The TSI is obtained by placing the pixel strips of the frames on the counting line in chronological order. An example of a counting line on a few consecutive frames and the corresponding TSI are shown in Figs. 6*a* and *b*.

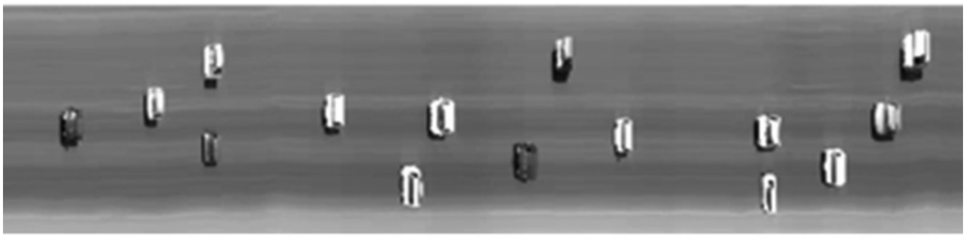
Each vehicle passing the counting line corresponds to an object in the TSI. The horizontal length of an object in a TSI corresponds to the number of frames in which the vehicle is on the counting line. As we apply the algorithm for highway traffic monitoring, it will give us information about the vehicle length if we suppose that the passing vehicle velocity changes are insignificant [27].

When the vehicle counting algorithm counts a new vehicle, we get its bounding box and the frame number. Then we can find the corresponding blob in the TSI for a bounding box with the *Y*-axis value of the top-left corner and the *Y*-axis value of the top-right corner that appear in a special frame (Fig. 7*a*). We keep only this strip and remove all unimportant parts of the TSI.

To find the end of the vehicle, first, we apply the Canny edge detector to the complete TSI. Second, we erode the result with a



a



b

Fig. 6 *Counting line on a few consecutive frames and the corresponding TSI*
a Vehicle counting line (white line) on a few consecutive frames
b TSI generated from the vehicle counting line wherein each object corresponds to a vehicle

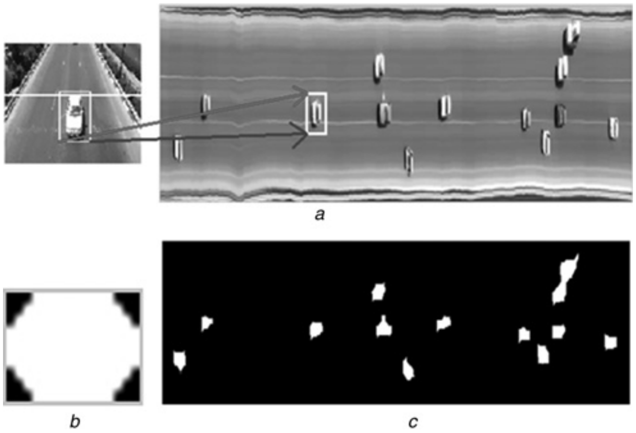


Fig. 7 *Using the position of vehicle in video frame to find its corresponding blob position in the TSI*

a Finding a vehiclès corresponding blob in the TSI
b Structuring element which is used for the close operation
c TSI obtained after morphological operation

vertical structuring element of sizes (2, 1) to eliminate the adverse effects of cracked asphalt and the guiding lines on the road. Then we close the result with an ellipsoid structuring element which is given in Fig. 9*b*. The result will contain some blobs (Fig. 9*c*).

Each blob corresponds to a vehicle. Considering the frame number in which the vehicle is counted on the counting line, we move on to the obtained binary TSI to the right till we reach the end of the blob, where there are a small number of white pixels. Therefore, the end of the vehicle is obtained and the vehicle length can be simply computed. Matching the vehicle location in the frame and the corresponding blob location in the TSI gives us good advantages. Since TSI is obtained after some morphological operations, it is sensitive to noise. In addition, it is probable that two vehicle blobs which are moving close to each other join together and make one blob. When we extract the exact location of the vehicle blob in the TSI, the algorithm uses exact information. The only problem that can occur is reporting a wrong vehicle length which results from joining the blobs of two very close vehicles. This problem may occur in a very congested traffic condition. However, solving the problem in such conditions is not the main goal of this paper. The length of the vehicle is a valuable feature, but as we extract it from TSI, it is sensitive to the velocity of the vehicle since this length is actually equal to the number of frames that the vehicle remains on the counting line. In addition, it is still sensitive to the used morphological operation which can reduce the accuracy of the results. However, we use it as it provides valuable information for vehicle classification; and we add a texture-based feature. This feature is explained below.

3.3.2. Correlation of GLCM: GLCM can be used to compute a number of texture-based features of an image. To do this, the bounding boxes of the counted vehicles which are extracted automatically in the counting procedure and are passed to the classification part are used. We apply the Canny edge detector to the bounding boxes. Then, we make GLCM from the result and calculate the correlation value from them. This value is greater for large vehicles like buses and trucks and becomes smaller as the size of the vehicle becomes smaller.

Finally, after extracting the above two features, we train a RF using a small number of training data and use it for categorisation. The RF grows many classification trees. To classify a new vehicle from its features, put the features down each of the trees in the forest. Each tree votes for a class. The forest chooses the class having the most votes (over all the trees in the forest). As the RF has inconsiderable run-time and computational load, we can use it in real-time applications.

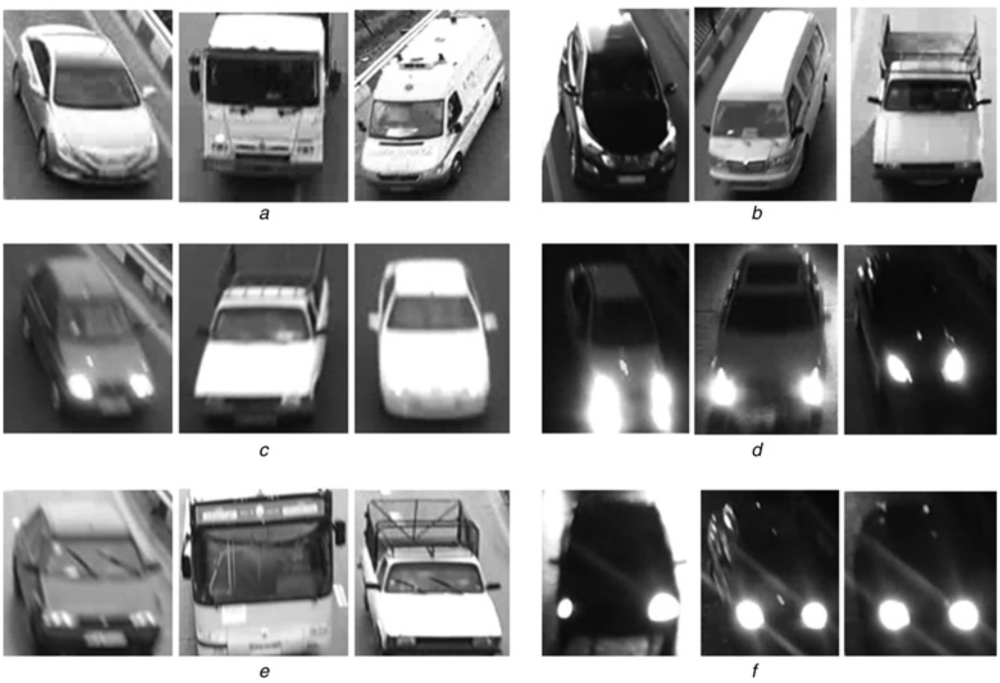


Fig. 8 *Sample training image at*

a Noon
b Afternoon
c Sundown
d Night
e Rainy day
f Rainy night

IET Intell. Transp. Syst., 2016, Vol. 10, Iss. 6, pp. 406–413

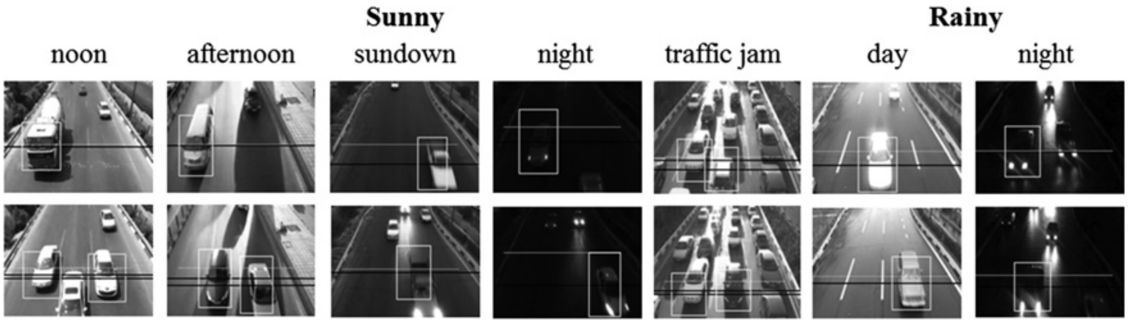


Fig. 9 *Some detected and counted vehicles which are passing the counting line in the proposed methods. White lines represent the counting lines. Black lines represent the middle of vehicle bounding boxes along the x-axis which have passed the counting line, resulting in the counting of the vehicles*

4 Experiments

To evaluate the proposed method, we collected a dataset from some urban highways with smooth traffic of different types of vehicles. The camera is mounted on a bridge about 6 m high over the highway. Videos are recorded from the top and front views of the vehicles. The common structure of the vehicles windshield and the absence of disturbing effects, especially on the road side, are the reasons why we chose this view. Common challenges in the highways such as various lighting conditions, variable weather conditions, and also camera vibration or image blurring are all considered in each video stream. We have used four videos taken at noon, afternoon, sundown and night on a sunny day, two videos on a rainy day and night, with about 5 min long for each one and a 1 min video from traffic jam to evaluate the algorithm performance in various lighting and weather conditions. We produced videos at 30 frames per second. This dataset is available for public use on Amirkabir Computer Vision Laboratory Website (http://www.cvlab.aut.ac.ir/?q=node/27). The algorithms are implemented and executed in MATLAB R2009b and C++ in a windows 7 operational system with 32GB RAM and a 3.6 GHz processor.

The active basis is formed by 50 Gabor wavelet elements. The common sketch is learned from the training images considering different types of vehicles extracted from videos in various lighting and weather conditions (Figs. 8*a–f*).

We set the common sketch similarity threshold between 30 and 50 experimentally. The symmetry threshold is considered as 0.3. As we mentioned before, the algorithm is able to detect vehicles in about the middle three-fifths of the frame. Vehicles in the videos come towards the camera. Even for fast moving vehicles, there are a sufficient number of frames that capture the vehicles in about the middle three-fifths of the frames. Thus, the above assumptions do not have considerably bad effects on the detection performance. First, we compared the results of the proposed method with [22] results given in Table 1. The bold values show the best results for each case. We applied the methods on traffic images extracted from a video dataset at various lighting conditions. Method [22] uses colour symmetry in addition to edge symmetry in order to verify the vehicles. It leads to low accuracy because vehicles in different lighting conditions considering shadow and sunlight are not symmetrical in colour. We obtained better results by ignoring colour symmetry.

Table 2 shows the vehicle detection results for the proposed method and those of [30, 12] on the video dataset. To gain higher accuracy in this step, we blur each detected vehicle with an average filter and apply the vehicle detection algorithm to the changed frame. We repeat this until it cannot detect a vehicle candidate with a similarity value higher than a threshold. Therefore, all vehicles can be detected in the videos.

Method [30] is based on the TSI generation and blob extracting using morphology. The morphology is not able to differentiate the blobs correctly in some cases especially in the night because of vehicle moving headlight and the afternoon because of long vehicle shadow. Method [12] uses a histogram-based background subtraction method to detect foreground objects. It is not able to give good accuracy at sundown and night which do not have rich histogram information. In comparison to the other mentioned methods, the proposed approach can detect vehicles and shadows or background changes have no adverse effect on the algorithm performance (Fig. 9). In addition, the algorithm gives exact information about the detected vehicle position.

Table 3 compares the counting results for the proposed method and the method of [30] which counts all blobs of TSI as the vehicles number. It is worth mentioning that the blobs are extracted using morphology operation and are error prone.

The proposed algorithm never counts a vehicle twice; but, it may miss some vehicles because of the assumptions we made in the detection algorithm. As we use only one frame of each five consecutive frames for counting, the detection algorithm may detect some vehicles in the frames which are ignored in counting. These uncounted vehicles are actually detected in the processed frames only before the counting line or only after it. Here, we see a trade-off between the counting algorithm accuracy and the detection algorithm speed.

Counted vehicles are given to the classification step. The method we used to find the blob corresponding to a vehicle in the TSI allows us to extract the length feature more exactly. Adding the GLCM correlation also helps the classifier to give better results. Finally, a

Table 1 Vehicle detection results on traffic images

<table>
<tr><th></th><th></th><th colspan="4">Sunny day</th></tr>
<tr><th></th><th></th><th>Noon</th><th>Afternoon</th><th>Sundown</th><th>Night</th></tr>
<tr><td>actual number of vehicles</td><td></td><td>228</td><td>413</td><td>261</td><td>277</td></tr>
<tr><td>total learning time, s</td><td>Yao et al. [22]</td><td colspan="4">2.5</td></tr>
<tr><td></td><td>our method</td><td colspan="4">2.5</td></tr>
<tr><td>percentage of correctly detected vehicles</td><td>Yao et al. [22]</td><td>64.47</td><td>70.70</td><td>81.23</td><td>77.62</td></tr>
<tr><td></td><td>our method</td><td>92.11</td><td>74.82</td><td>85.06</td><td>84.12</td></tr>
<tr><td>percentage of falsely detected vehicles</td><td>Yao et al. [22]</td><td>0.004</td><td>0.007</td><td>0.008</td><td>0.040</td></tr>
<tr><td></td><td>our method</td><td>0.008</td><td>0.012</td><td>0.004</td><td>0.032</td></tr>
<tr><td>average run-time per frame, s</td><td>Yao et al. [22]</td><td colspan="4">0.11</td></tr>
<tr><td></td><td>our method</td><td colspan="4">0.1</td></tr>
</table>

Table 2 Vehicle detection results on traffic videos

		Sunny day					Rainy day	
		Noon	Afternoon	Sundown	Night	Traffic jam	Day	Night
actual number of vehicle		228	413	261	277	72	128	247
total learning time, s	Rashid *et al.* [30]				–			
	Wu and Juang [12]				–			
	our method				2.5			
percentage of correctly detected vehicle	Rashid *et al.* [30]	86.8	48.4	82.8	30.7	50.0	50	43.3
	Wu and Juang [12]	94.7	46.0	0.7	6.14	13.88	55.5	15.4
	our method	**99.1**	**99.0**	**98.4**	**99.3**	**97.2**	**94.5**	**98.8**
percentage of falsely detected vehicle	Rashid *et al.* [30]	182.9	146.5	84.7	141.2	222.8	84.4	72.1
	Wu and Juang [12]	11.84	5.1	**1.9**	3.6	**2.7**	24.2	6.9
	our method	**2.2**	**1.9**	2.3	**2.9**	41.1	**4.7**	**3.6**
average run-time per frame, s	Rashid *et al.* [30]				**0.15**			
	Wu and Juang [12]				0.4			
	our method				0.2			

Table 3 Vehicle counting results

		Sunny day					Rainy day	
		Noon	Afternoon	Sundown	Night	Traffic jam	Day	Night
actual number of vehicle		228	413	261	277	72	128	247
percentage of correctly counted vehicle	Rashid *et al.* [30]	86.8	48.4	**82.8**	30.7	50.0	50	43.3
	our method	**90.5**	**48.7**	68.2	**66.8**	**63.8**	**51.6**	**46.6**
percentage of falsely counted vehicle	Rashid *et al.* [30]	182.9	146.5	84.7	141.2	222.8	84.4	72.1
	our method	**7.9**	**2.9**	**4.2**	**21.7**	**11.11**	**4.7**	**1.6**
average run-time per frame, s	Rashid *et al.* [30]				**0.000004**			
	our method				0.0045			

RF consisting of 100 trees is made and trained with 10 training data. We compare our method with methods [27, 12] given in Table 4.

The confusion matrices are given in Table 5. Method [27] benefits from blob shape-based features like blob length, width and centroid for classification and method [12] classify vehicles according to their size. They are both not able to give good accuracy in real traffic situations. Especially with the presence of shadow and in variable lighting conditions.

Table 4 Vehicle classification results

		Sunny day					Rainy day	
		Noon	Afternoon	Sundown	Night	Traffic jam	Day	Night
actual number of vehicle		228	413	261	277	72	128	247
s: number of small vehicles		*s*: 214	*s*: 394	*s*: 253	*s*: 264	*s*: 259	*s*: 122	*s*: 235
m: number of medium vehicles		*m*: 11	*m*: 17	*m*: 8	*m*: 11	*m*: 4	*m*: 5	*m*: 11
l: number of large vehicles		*l*: 3	*l*: 2	*l*: 0	*l*: 2	*l*: 1	*l*: 1	*l*: 1
percentage of successfully classified vehicles	Mithun *et al.* [27]	25.0	13.1	20.7	23.9	10.1	10.2	23.1
	Wu and Juang [12]	64.4	28.6	0.8	5.1	5.8	26.6	11.3
	our method	**71.9**	**36.6**	**59.8**	**44.0**	**51.3**	**53.1**	**44.9**
learning time, s	Mithun *et al.* [27]				0.000001			
	Wu and Juang [12]				–			
	our method				0.0006			
average run-time per frame, s	Mithun *et al.* [27]				0.0002			
	Wu and Juang [12]				**0.00002**			
	our method				0.0001			

Table 5 Classification confusion matrix. The first, second and third rows in each cell report classification results obtained from [27, 12] and the proposed methods, respectively

Noon				Afternoon			Sundown			Night			Traffic Jam			Rainy day			Rainy night		
	s	*m*	*l*	*s*	*m*	*l*	*s*	*m*	*l*	*s*	*m*	*l*	*s*	*m*	*l*	*s*	*m*	*l*	*s*	*m*	*l*
s	52	51	109	54	164	193	53	114	63	66	98	132	7	0	29	11	13	36	57	13	33
	144	0	63	117	0	66	2	0	0	14	0	3	4	0	5	32	0	37	28	0	10
	153	40	6	**150**	0	44	**156**	0	24	**122**	0	70	**43**	4	2	**67**	2	1	**111**	4	2
m	1	2	5	0	**0**	3	0	**1**	1	0	**0**	1	0	0	0	0	**1**	0	1	**1**	1
	1	0	5	1	**0**	5	0	0	0	0	**0**	0	0	0	0	0	0	1	0	0	0
	1	**7**	1	6	**0**	2	6	0	1	4	**0**	3	0	**1**	2	1	0	0	1	0	0
l	0	1	3	0	1	0	0	0	**0**	0	0	**0**	0	0	0	0	0	**1**	0	1	**0**
	0	0	3	0	0	**1**	0	0	**0**	0	0	**0**	0	0	0	0	0	**1**	0	0	**0**
	0	0	**4**	1	0	**1**	0	0	**0**	0	0	**0**	0	0	**1**	0	0	**1**	0	0	**0**

Totally speaking, the proposed method outperforms similar methods because of its accuracy and ability to handle real outdoor conditions with various lighting and weather conditions in smooth traffic. In traffic jams, although the proposed method gives better results than other similar methods; the results still need to be improved with further research. In addition, the proposed vehicle detection and counting methods have several parameters which are presently determined experimentally. It is desired to have the algorithm estimate these parameters adaptively in order to reduce the user interference. Finally, the proposed method, similar to most of the existing methods, is view-dependent. In order to use it with cameras at different views, modifications are needed to reduce the algorithm sensitivity to views. A view-independent traffic monitoring system is highly desirable.

5 Conclusions

In this paper, we proposed algorithms for vehicle detection, counting and classification for highway traffic monitoring systems. We used ABM to find vehicle candidates in the video frames and verified them by checking the reflection symmetry. Vehicles detected in sequential frames are passed to the counting procedure and vehicles counted are given to the classification step. The vehicle length in the TSI image and correlation value calculated from the GLCM matrix are used to train a RF and classify vehicles into three groups: small (e.g. car), medium (e.g. van) and large (e.g. bus and trunk) vehicles. Experimental results show the high performance of the proposed method and its good accuracy in real environments with common challenges existing in highways such as various lighting and weather conditions, shadow, camera vibration and image blurring. Some improvements such as using GPU programming or adding other texture-based features to the classification part are suggested to improve the algorithm performance and run-time.

6 References

1 Wu, K., Xu, T., Zhang, H.: 'Overview of video-based vehicle detection technologies'. 6th Int. Conf. on Computer Science & Education (ICCSE), August 2011, pp. 821–825

2 Atiq, H.M., Farooq, U., Ibrahim, R., *et al.*: 'Vehicle detection and shape recognition using optical sensors: a review'. Second Int. Conf. on Machine Learning and Computing (ICMLC), Bangalore, February 2010, pp. 223–227

3 Liu, Y., Tian, B., Chen, S., *et al.*: 'A survey of vision-based vehicle detection and tracking techniques in ITS'. IEEE Int. Conf. on Vehicular Electronics and Safety (ICVES), Dongguan, July 2013, pp. 72–77

4 Jalali Moghaddam, M., Hosseini, M., Safabakhsh, R.: 'Traffic light control based on Fuzzy Q_learning'. Int. Symp. on Artificial Intelligence and Signal Processing (AISP), Mashhad, March 2015, pp. 124–128

5 Kalaki, A.S., Safabakhsh, R.: 'Current and adjacent lanes detection for an autonomous vehicle to facilitate obstacle avoidance using a monocular camera'. Iranian Conf. on Intelligent Systems (ICIS), Bam, February 2014, pp. 1–6

6 Tsai, L.W., Hsieh, J.W., Fan, K.C.: 'Vehicle detection using normalized color and edge map', *IEEE Trans. Image Process.*, 2007, **16**, (3), pp. 850–864

7 Ghasemi, A., Safabakhsh, R.: 'A real-time multiple vehicle classification and tracking system with occlusion handling'. IEEE Int. Conf. on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, August 2012, pp. 109–115

8 Mandellos, N.A., Keramitsoglou, I., Kiranoudis, C.T.: 'A background subtraction algorithm for detecting and tracking vehicles', *Expert Syst. Appl.*, 2011, **38**, (3), pp. 1619–1631

9 Unzueta, L., Nieto, M., Cortés, A., *et al.*: 'Adaptive multicue background subtraction for robust vehicle counting and classification', *IEEE Trans. Intell. Transp. Syst.*, 2012, **13**, (2), pp. 527–540

10 Pang, C.C.C., Lam, W.W.L., Yung, N.H.C.: 'A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images', *IEEE Trans. Intell. Transp.*, 2007, **8**, (3), pp. 441–459

11 Zhou, J., Gao, D., Zhang, D.: 'Moving vehicle detection for automatic traffic monitoring', *IEEE Trans. Veh. Technol.*, 2007, **56**, (1), pp. 51–59

12 Wu, B.F., Juang, J.H.: 'Adaptive vehicle detector approach for complex environments', *IEEE Trans. Intell. Transp. Syst.*, 2012, **13**, (2), pp. 817–827

13 Gangodkar, D., Kumar, P., Mittal, A.: 'Robust segmentation of moving vehicles under complex outdoor conditions', *IEEE Trans. Intell. Transp. Syst.*, 2012, **13**, (4), pp. 1738–1752

14 Chen, Z., Ellis, T., Velastin, S.A.: 'Vehicle detection, tracking and classification in urban traffic'. 15th Int. IEEE Conf. on Intelligent Transportation Systems (ITSC), Anchorage, September 2012, pp. 951–956

15 Jia, Y., Zhang, C.: 'Front-view vehicle detection by Markov chain Monte Carlo method', *Pattern Recognit.*, 2009, **42**, (3), pp. 313–321

16 Sutar, V.B., Admuthe, M.L.: 'Night time vehicle detection and classification using support vector machine', *IOSR J. VLSI Signal Process. (IOSR-JVSP)*, 2012, **1**, (4), pp. 1–9

17 Chen, Y.L., Wu, B.F., Huang, H.Y.: 'A real-time vision system for night time vehicle detection and traffic surveillance', *IEEE Trans. Ind. Electron.*, 2011, **58**, (5), pp. 2030–2044

18 Leon, L.C., Hirata, R.: 'Vehicle detection using mixture of deformable parts models: Static and dynamic camera'. 25th SIBGRAPI Conf. on Graphics, Patterns and Images (SIBGRAPI), Ouro Preto, August 2012, pp. 237–244

19 Li, Y., Wang, F.Y., Li, B., *et al.*: 'A multi-scale model integrating multiple features for vehicle detection'. 16th Int. IEEE Conf. on Intelligent Transportation Systems-(ITSC), The Hague, October 2013, pp. 399–403

20 Li, Y., Li, B., Tian, B., *et al.*: 'Vehicle detection based on the AND–OR graph for congested traffic conditions', *IEEE Trans. Intell. Transp. Syst.*, 2013, **14**, (2), pp. 984–993

21 Li, Y., Tian, B., Li, B., *et al.*: 'Vehicle detection with a part-based model for complex traffic conditions'. IEEE Int. Conf. on Vehicular Electronics and Safety (ICVES), Dongguan, July 2013, pp. 110–113

22 Yao, Y., Xiong, G., Wang, K., *et al.*: 'Vehicle detection method based on active basis model and symmetry in ITS'. 16th Int. IEEE Conf. on Intelligent Transportation Systems-(ITSC), The Hague, October 2013, pp. 614–618

23 Li, Y., Yao, Q.: 'Rear lamp based vehicle detection and tracking for complex traffic conditions'. 9th IEEE Int. Conf. on Networking, Sensing and Control (ICNSC), Beijing, April 2012, pp. 387–392

24 Chiu, C.C., Ku, M.Y., Wang, C.Y.: 'Automatic traffic surveillance system for vision-based vehicle recognition and tracking', *J. Inf. Sci. Eng.*, 2012, **26**, (2), pp. 611–629

25 Li, S., Yu, H., Zhang, J., *et al.*: 'Video-based traffic data collection system for multiple vehicle types', *IET Intell. Transp. Syst.*, 2013, **8**, (2), pp. 164–174

26 Huang, D.Y., Chen, C.H., Hu, W.C., *et al.*: 'Feature-based vehicle flow analysis and measurement for a real-time traffic surveillance system', *J. Inf. Hiding Multimedia Signal Process.*, 2012, **3**, (3), pp. 279–294

27 Mithun, N.C., Rashid, N.U., Rahman, S.M.: 'Detection and classification of vehicles from video using multiple time-spatial images', *IEEE Trans. Intell. Transp. Syst.*, 2012, **13**, (3), pp. 1215–1225

28 Yang, M.T., Jhang, R.K., Hou, J.S.: 'Traffic flow estimation and vehicle-type classification using vision-based spatial–temporal profile analysis', *IET Comput. Vis.*, 2013, **7**, (5), pp. 394–404

29 Wu, Y.N., Si, Z., Gong, H., *et al.*: 'Learning active basis model for object detection and recognition', *Int. J. Comput. Vis.*, 2010, **90**, (2), pp. 198–235

30 Rashid, N.U., Mithun, N.C., Joy, B.R., *et al.*: 'Detection and classification of vehicles from a video using time-spatial image'. Int. Conf. on Electrical and Computer Engineering (ICECE), Dhaka, December 2010, pp. 502–505