European Transport Research Review

ORIGINAL PAPER Open Access

Vision-based vehicle detection and counting system using deep learning in highway scenes

Huansheng Song, Haoxiang Liang*, Huaiyu Li, Zhe Dai and Xu Yun

Abstract

Intelligent vehicle detection and counting are becoming increasingly important in the field of highway management. However, due to the different sizes of vehicles, their detection remains a challenge that directly affects the accuracy of vehicle counts. To address this issue, this paper proposes a vision-based vehicle detection and counting system. A new high definition highway vehicle dataset with a total of 57,290 annotated instances in 11,129 images is published in this study. Compared with the existing public datasets, the proposed dataset contains annotated tiny objects in the image, which provides the complete data foundation for vehicle detection based on deep learning. In the proposed vehicle detection and counting system, the highway road surface in the image is first extracted and divided into a remote area and a proximal area by a newly proposed segmentation method; the method is crucial for improving vehicle detection. Then, the above two areas are placed into the YOLOv3 network to detect the type and location of the vehicle. Finally, the vehicle trajectories are obtained by the ORB algorithm, which can be used to judge the driving direction of the vehicle and obtain the number of different vehicles. Several highway surveillance videos based on different scenes are used to verify the proposed methods. The experimental results verify that using the proposed segmentation method can provide higher detection accuracy, especially for the detection of small vehicle objects. Moreover, the novel strategy described in this article performs notably well in judging driving direction and counting vehicles. This paper has general practical significance for the management and control of highway scenes.

Keywords: Vehicle dataset, Image segmentation, Vehicle detection, Vehicle counting, Highway management

Introduction

Vehicle detection and statistics in highway monitoring video scenes are of considerable significance to intelligent traffic management and control of the highway. With the popular installation of traffic surveillance cameras, a vast database of traffic video footage has been obtained for analysis. Generally, at a high viewing angle, a more-distant road surface can be considered. The object size of the vehicle changes greatly at this viewing angle, and the detection accuracy of a small object far away from the road is low. In the face of complex camera scenes, it is essential to effectively solve the above problems and further apply them. In this article, we focus on the above issues to propose a viable solution, and we apply the vehicle detection results to multi-object tracking and vehicle counting.

*Correspondence: lianghx7@gmail.com
School of Information Engineering, Chang'an University, Middle Section of Nan Erhuan Road, Xi'an, China

Related work on vehicle detection

At present, vision-based vehicle object detection is divided into traditional machine vision methods and complex deep learning methods. Traditional machine vision methods use the motion of a vehicle to separate it from a fixed background image. This method can be divided into three categories [1]: the method of using background subtraction [2], the method of using continuous video frame difference [3], and the method of using optical flow [4]. Using the video frame difference method, the variance is calculated according to the pixel values of two or three consecutive video frames. Moreover, the moving foreground region is separated by the threshold [3]. By using this method and suppressing noise, the stopping of the vehicle can also be detected [5]. When the

background image in the video is fixed, the background information is used to establish the background model [5]. Then, each frame image is compared with the background model, and the moving object can also be segmented. The method of using optical flow can detect the motion region in the video. The generated optical flow field represents each pixel's direction of motion and pixel speed [4]. Vehicle detection methods using vehicle features, such as the Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) methods, have been widely used. For example, 3D models have been used to complete vehicle detection and classification tasks [6]. Using the correlation curves of 3D ridges on the outer surface of the vehicle [7], the vehicles are divided into three categories: cars, SUVs, and minibuses.

The use of deep convolutional networks (CNNs) has achieved amazing success in the field of vehicle object detection. CNNs have a strong ability to learn image features and can perform multiple related tasks, such as classification and bounding box regression [8]. The detection method can be generally divided into two categories. The two-stage method generates a candidate box of the object via various algorithms and then classifies the object by a convolutional neural network. The one-stage method does not generate a candidate box but directly converts the positioning problem of the object bounding box into a regression problem for processing. In the two-stage method, Region-CNN (R-CNN) [9] uses selective region search [10] in the image. The image input to the convolutional network must be fixed-size, and the deeper structure of the network requires a long training time and consumes a large amount of storage memory. Drawing on the idea of spatial pyramid matching, SPP NET [11] allows the network to input images of various sizes and to have fixed outputs. R-FCN, FPN, and Mask RCNN have improved the feature extraction methods, feature selection, and classification capabilities of convolutional networks in different ways. Among the one-stage methods, the most important are the Single Shot Multibox Detector (SSD) [12] and You Only Look Once (YOLO) [13] frameworks. The MutiBox [14], Region Proposal Network (RPN) and multi-scale representation methods are used in SSD, which uses a default set of anchor boxes with different aspect ratios to more accurately position the object. Unlike SSD, the YOLO [13] network divides the image into a fixed number of grids. Each grid is responsible for predicting objects whose centre points are within the grid. YOLOv2 [15] added the BN (Batch Normalization) layer, which makes the network normalize the input of each layer and accelerate the network convergence speed. YOLOv2 uses a multi-scale training method to randomly select a new image size for every ten batches. Our vehicle object detection uses the YOLOv3 [16] network. Based on YOLOv2, YOLOv3 uses logistic regression for the object category. The category loss method is two-class cross-entropy loss, which can handle multiple label problems for the same object. Moreover, logistic regression is used to regress the box confidence to determine if the IOU of the a priori box and the actual box is greater than 0.5. If more than one priority box satisfies the condition, only the largest prior box of the IOU is taken. In the final object prediction, YOLOv3 uses three different scales to predict the object in the image.

The traditional machine vision method has a faster speed when detecting the vehicle but does not produce a good result when the image changes in brightness, there is periodic motion in the background, and where there are slow moving vehicles or complex scenes. Advanced CNN has achieved good results in object detection; however, CNN is sensitive to scale changes in object detection [17, 18]. The one stage method uses grids to predict objects, and the grid's spatial constraints make it impossible to have higher precision with the two-stage approach, especially for small objects. The two stage method uses region of interest pooling to segment candidate regions into blocks according to given parameters, and if the candidate region is smaller than the size of the given parameters, the candidate region is padded to the size of the given parameters. In this way, the characteristic structure of a small object is destroyed and its detection accuracy is low. The existing methods do not distinguish if large and small objects belong to the same category. The same method is used to deal with the same type of object, which will also lead to inaccurate detection. The use of image pyramids or multi-scale input images can solve the above problems, although the calculation requirements are large.

Vehicle detection research in Europe

Vision-based vehicle detection methods in Europe have achieved abundant results. In [19], between the "Hofolding" and "Weyern" sections of the A8 motorway in Munich, Germany, the Multivariate Alteration Detection (MAD) method [20] was used to detect the change of two images with a short time lag. The moving vehicles are highlighted in a change image, which is used to estimate the vehicle density of the road. In [21], using the motorways A95 and A96 near Munich, the A4 near Dresden, and the "Mittlere Ring" in Munich as the test environments, the Canny edge algorithm [22] is applied to the road image, and the histogram of the edge steepness is calculated. Then, using the k-means algorithm, the edge steepness statistics are divided into three parts, and a closed vehicle model is detected based on the steepness. A contrast-based approach was used to create a colour model to identify and remove vehicle shadow areas [23], which eliminates interference caused by movement in the scene. After eliminating the shadow area, the vehicle detection performance can be significantly improved. The

experiment in [23] was conducted on Italian and French highways. The HOG and Haar-like features were compared in [24], and the two features were merged to construct a detector for vehicle detection that was tested on French vehicle images. However, when the above method is used for vehicle detection, the type of vehicle cannot be detected. Additionally, when the illumination is insufficient, it is difficult to extract the edge of the vehicle or detect the moving car, which causes problems in low vehicle detection accuracy and affects the detection results for further use. Pictures of aerial view angles were used by [19, 20] but cannot clearly capture the characteristics of each car and produce false vehicle detections.

Nonetheless, with the development of deep learning technology, vehicle detection based on CNN has been successfully applied in Europe. In [25], Fast R-CNN was used for vehicle detection in traffic scenes in the city of Karlsruhe, Germany. Fast R-CNN uses a selective search strategy to find all candidate frames, which is notably time-consuming, and the vehicle detection speed is slow.

In short, research on vision-based vehicle detection is still progressing, and major challenges are gradually being overcome, which will make a significant contribution to the development of European traffic construction.

Related work on vehicle tracking

Advanced vehicle object detection applications, such as multi-object tracking, are also a critical ITS task [26]. Most multi-object tracking methods use Detection-Based Tracking (DBT) and Detection-Free Tracking (DFT) for object initialization. The DBT method uses background modeling to detect moving objects in video frames before tracking. The DFT method needs to initialize the object of the tracking but cannot handle the addition of new objects and the departure of old objects. The Multiple Object Tracking algorithm needs to consider the similarity of intra-frame objects and the associated problem of inter-frame objects. The similarity of intra-frame objects can use normalized cross-correlation (NCC). The Bhattacharyya distance is used to calculate the distance of the colour histogram between the objects, such as in [27]. When inter-frame objects are associated, it is necessary to determine that an object can only appear on one track and that one track can only correspond to one object. Currently, detection-level exclusion or trajectory-level exclusion can solve this problem. To solve the problems caused by scale changes and illumination changes of moving objects, [28] used SIFT feature points for object tracking, although this is slow. The ORB feature point detection algorithm [29] is proposed for use in this work. ORB can obtain better extraction feature points at a significantly higher speed than SIFT.

In summary, it can be considered that the method of vehicle object detection has been transferred from research on traditional methods to that on deep convolutional network methods. Moreover, there are fewer public datasets for specific traffic scenes. The sensitivity of convolutional neural networks to scale changes makes small object detection inaccurate. It is challenging to conduct multi-object tracking and subsequent traffic analysis when highway surveillance cameras are used. In summary, our contributions include the following:

1. A large-scale high definition dataset of highway vehicles is established that can provide many different vehicle objects fully annotated under various scenes captured by highway surveillance cameras. The dataset can be used to evaluate the performance of many vehicle detection algorithms when dealing with vehicle scale changes.
2. A method for detecting small objects in highway scenes is used to improve vehicle detection accuracy. The highway road surface area is extracted and divided into the remote area and the proximal area, which are placed into the convolution network for vehicle detection.
3. A multi-object tracking and trajectory analysis method is proposed for highway scenes. The detection object feature points are extracted and matched by the ORB algorithm, and the road detection line is determined to count the vehicle movement direction and the traffic flow.

This research will be described in the following sections. "Vehicle dataset" section introduces the vehicle dataset used in this paper. "The system structure" section introduces the overall process of the proposed system. "Methods" section describes our strategy in detail. "Results and discussion" section presents the experiments and related analysis. "Conclusions" section summarizes the entire article.

Vehicle dataset

Surveillance cameras in roads have been widely installed worldwide but traffic images are rarely released publicly due to copyright, privacy, and security issues. From the image acquisition point of view, the traffic image dataset can be divided into three categories: images taken by the car camera, images taken by the surveillance camera, and images taken by non-monitoring cameras [30]. The KITTI benchmark dataset [31] contains images of highway scenes and ordinary road scenes used for automatic vehicle driving and can solve problems such as 3D object detection and tracking. The Tsinghua-Tencent Traffic-Sign Dataset [32] has 100,000 images from car cameras covering various lighting conditions and weather conditions, although no vehicles are marked. The Stanford Car Dataset [33] is a vehicle dataset taken by non-monitoring cameras with a bright vehicle appearance. This dataset includes 19,618 categories of vehicles covering the brands, models, and production years of the vehicles. The Comprehensive Cars Dataset [34] is similar to the Stanford Car

Dataset but contains many pictures. The 27,618 images include the vehicle's maximum speed, number of doors, number of seats, displacement, and car type. The 136,727 images include the overall appearance of the vehicle. The datasets are taken by surveillance cameras; an example is the BIT-Vehicle Dataset [35], which contains 9,850 images. This dataset divides the vehicle into six types: SUV, sedan, minivan, truck, bus, and micro-bus; however, the shooting angle is positive, and the vehicle object is too small for each image, which is difficult to generalize for CNN training. The Traffic and Congestions (TRANCOS) dataset [36] contains pictures of vehicles on highways captured by surveillance cameras and contains a total of 1,244 images. Most of the images have some occlusion. This dataset has a small number of pictures, and no vehicle type is provided, which makes it less applicable. Therefore, few datasets have useful annotations, and few images are available in traffic scenes.

This section introduces the vehicle dataset from the perspective of the highway surveillance video we produced. The dataset has been published in: http://drive.google.com/open?id=1li858elZvUgss8rC_yDsb5bDfiRyhdrX. The dataset picture is from the highway monitoring video of Hangzhou, China (Fig. 1). The highway monitoring camera was installed on the roadside and erected at 12 meters; it had an adjustable field of view and no preset position. The images from this perspective cover the far distance of the highway and contains vehicles with dramatic changes in scale. The dataset images were captured from 23 surveillance cameras for different scenes, different times, and different lighting conditions. This dataset divides the vehicles into three categories: cars, buses, and trucks (Fig. 2). The label file is stored in a text document that contains the numeric code of the object category and the normalized coordinate value of the bounding box. As shown in Table 1, this dataset has a total of 11,129 images and 57,290 annotation boxes.. The images have an RGB format and 1920*1080 resolution. Note that we annotated the smaller objects in the proximal road area, and the dataset thus contains vehicle objects with massive scale changes. An annotated instance near the camera has more features, and an instance far from the camera has few features. Annotated instances of different sizes are beneficial to the improvement of the detection accuracy of small vehicle objects. This dataset is divided into two parts: a training set and a test set. In our dataset, cars accounted for 42.17% , buses accounted for 7.74%, and trucks accounted for 50.09%. There are 5.15 annotated instances in each image on average. Figure 3 compares the difference between the number of annotated instances in our dataset and the PASCAL VOC, ImageNet, and COCO datasets. Our dataset is a universal dataset for vehicle targets that can be used in a variety of areas, such as Europe. Compared with the existing vehicle datasets,

Fig. 1 Scenes taken by multiple highway surveillance cameras. **a** Scene 1; **b** Scene 2; **c** Scene 3

Fig. 2 Vehicle labelling category of the dataset

our dataset has a large number of high definition images, sufficient lighting conditions, and complete annotations.

The system structure

This section describes the main structure of the vehicle detection and counting system. First, the video data of the traffic scene are entered. Then, the road surface area is extracted and divided. The YOLOv3 deep learning object detection method is used to detect the vehicle object in the highway traffic scene. Finally, ORB feature extraction is performed on the detected vehicle box to complete multi-object tracking and obtain vehicle traffic information.

According to Fig. 4, the road surface segmentation method is used to extract the road area of the highway. The road area is divided into two parts based on the position where the camera is erected, a remote area and a proximal area. Then, the vehicles in the two road areas are detected using the YOLOv3 object detection algorithm. This algorithm can improve the small object detection effect and solve the problem that the object is difficult to detect due to the sharp change of the object scale. The ORB algorithm is then used for multi-object tracking. The ORB algorithm extracts the detected box's features and matches them to achieve correlation between the same object and different video frames. Finally, traffic statistics are calculated. The trajectory generated by the object tracking is generated, the vehicle driving direction is determined, and traffic information such as the number of vehicles in each category is collected. This system improves the accuracy of object detection from the highway surveillance video perspective and constructs a detection tracking and traffic information acquisition plan within the full field of the camera view.

Methods

Road surface segmentation

This section describes the method of highway road surface extraction and segmentation. We implemented surface extraction and segmentation using image processing methods, such as Gaussian mixture modelling, which enables better vehicle detection results when using the deep learning object detection method. The highway surveillance video image has a large field of view. The vehicle is the focus of attention in this study, and the region of interest in the image is thus the highway road surface area. At the same time, according to the camera's shooting angle, the road surface area is concentrated in a specific range of the image. With this feature, we could extract the highway road surface areas in the video. The process of road surface extraction is shown in Fig. 5.

As shown in Fig. 5, to eliminate the influence of vehicles on the road area segmentation, we used the Gaussian mixture modeling method to extract the background in the first 500 frames of the video. The value of the pixel in the image is Gaussian around a certain central value in a certain time range, and each pixel in each frame of the image is counted. If the pixel is far from the centre, the pixel belongs to the foreground. If the value of the pixel point deviates from the centre value within a certain variance, the pixel point is considered to belong to the background. The mixed Gaussian model is especially useful in images where background pixels have multi-peak characteristics such as the highway surveillance images used in this study.

After extraction, the background image is smoothed by a Gaussian filter with a 3*3 kernel. The MeanShift algorithm is used to smooth the colour of the input image,

Table 1 Vehicle dataset information published in this study

Image format	Size	Total number of images	Total number of instances	$\frac{Totalnumberofinstances}{Totalnumberofimages}$
RGB	1920 x 1080	11,129	57,290	5.15

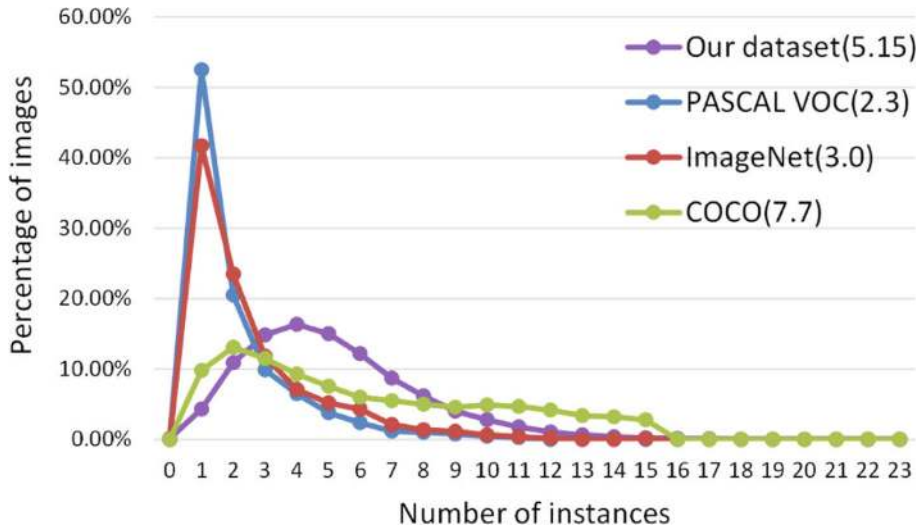


Fig. 3 Annotated instances per image (average numbers of annotated instances are shown in parentheses)

neutralize the colour with a similar colour distribution, and erode the colour area with a smaller area. On this basis, the flooding filling algorithm is used to separate the road surface area. The flooding filling algorithm selects a point in the road surface area as a seed point and fills the adjacent continuous road surface areas with the pixel value of the seed point. The pixel value of the adjacent continuous road surface areas is close to the seed point pixel value. Finally, the hole filling and morphological expansion operations are performed to more completely extract the road surface. We extracted the road surfaces of different highway scenes (Fig. 6), and the results are shown in Fig. 7.

We segmented the road surface area to provide accurate input for subsequent vehicle detection. For the extracted road surface image, a minimum circumscribed rectangle is generated for the image without rotation. The processed image is divided into five equal parts, the 1/5 area adjacent to the origin of the coordinate axis is defined as the near remote area of the road surface, and the remaining 4/5 area is defined as the near proximal area of the road surface. The near proximal area and the near remote area overlap by 100 pixels (as shown in the red part of Fig. 8) to address the problem that the vehicle in the image may be divided into two parts by the above procedure. The pixel values of the near proximal area and the near remote area are searched column by column. If the pixel values in the column are all zero, the image of the column is all black and is not the road surface area; it is then deleted. After the not-road-surface areas are excluded, the reserved areas are called remote areas and proximal areas of the road surface.

Vehicle detection using YOLOv3

This section describes the object detection methods used in this study. The implementation of the highway vehicle detection framework used the YOLOv3 network. The YOLOv3 algorithm continues the basic idea of the first two generations of YOLO algorithms. The convolutional neural network is used to extract the features

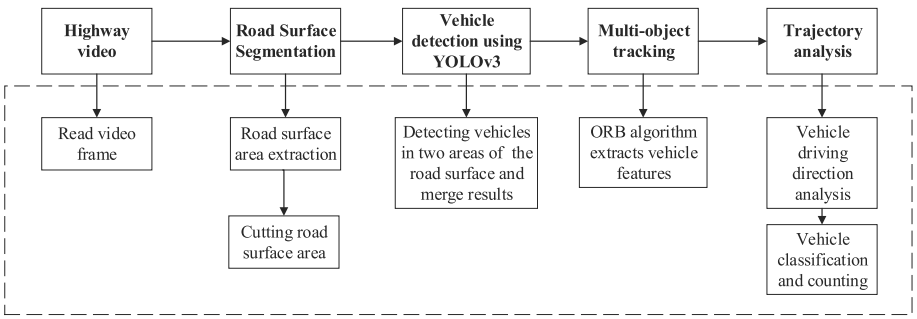


Fig. 4 Overall flow of the method

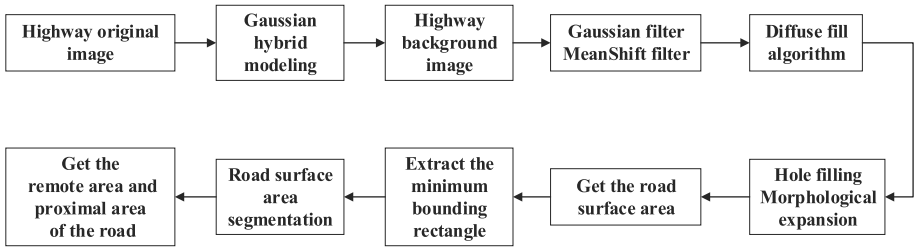


Fig. 5 Overall flow of road surface extraction

of the input image. According to the size of the feature map, such as 13*13, the input image is divided into 13*13 grids. The centre of the object label box is in a grid unit, and the grid unit is responsible for predicting the object. The network structure adopted by YOLOv3 is called Darknet-53. This structure adopts the full convolution method and replaces the previous version of the direct-connected convolutional neural network with the residual structure. The branch is used to directly connect the input to the deep layer of the network. Direct

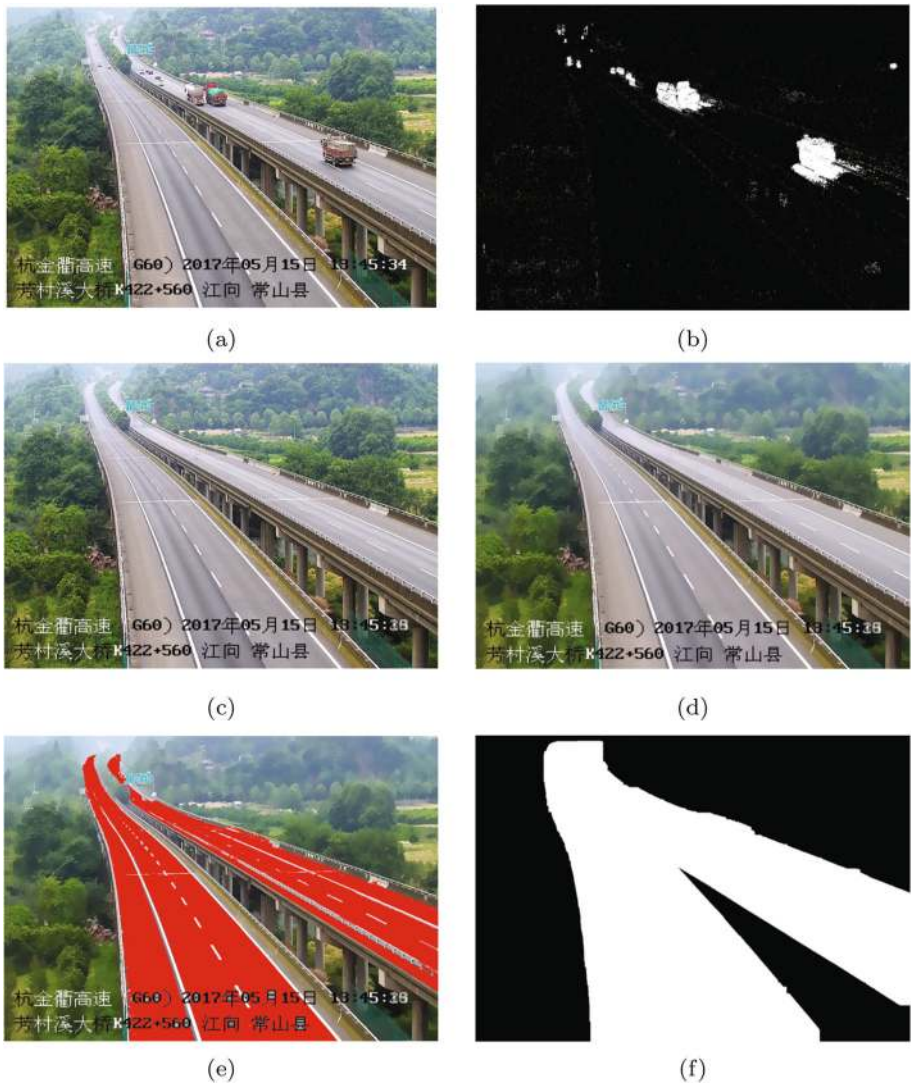


Fig. 6 Process of road surface area extraction. **a** Original image; **b** image foreground; **c** image background; **d** Gaussian filter and MeanShift filter; **e** diffuse filling; **f** road surface area mask

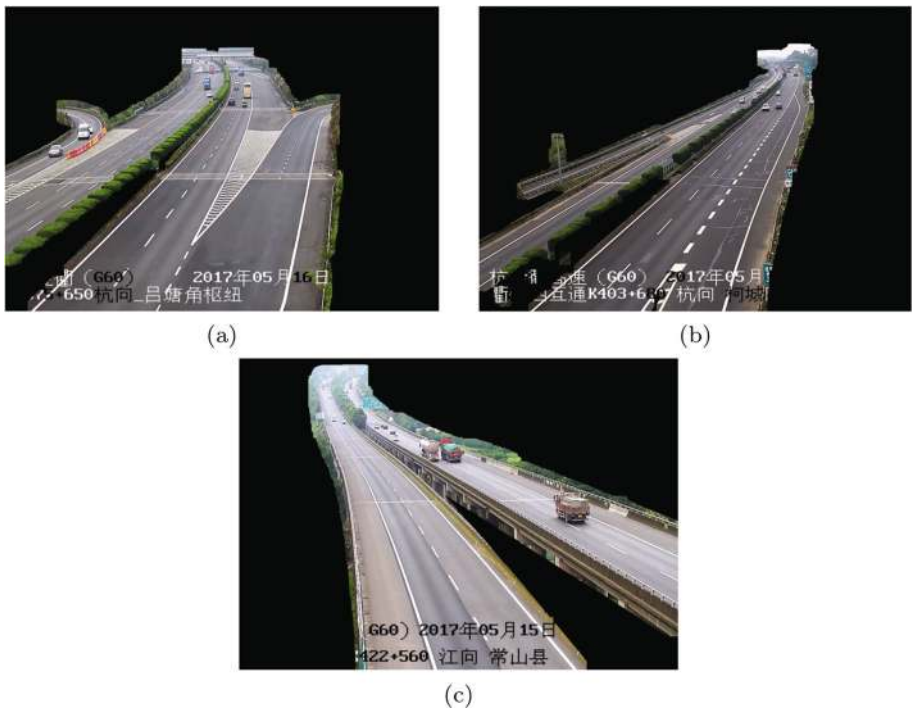


Fig. 7 Road surface extraction results for different highway scenarios. **a** Scene 1; **b** Scene 2; **c** Scene 3

learning of residuals ensures the integrity of image feature information, simplifies the complexity of training, and improves the overall detection accuracy of the network. In YOLOv3, each grid unit will have three bounding boxes of different scales for one object. The candidate box that has the largest overlapping area with the annotated box will be the final prediction result. Additionally, the YOLOv3 network has three output scales, and the three scale branches are eventually merged. Shallow features are used to detect small objects, and deep features are used to detect large objects; the network can thus detect objects with scale changes. The detection speed is fast, and the detection accuracy is high. Traffic scenes taken by highway surveillance video have good adaptability to the YOLOv3 network. The network will finally output the coordinates, confidence, and category of the object.

When using YOLO detection, images are resized to the same size, such as 416*416, when they are sent to the network. Since the image is segmented, the size of the remote road surface becomes deformed and larger. Therefore, more feature points of a small vehicle object can be acquired to avoid the loss of some object features due to the vehicle object being too small. The dataset presented in "Vehicle dataset" section is placed into the YOLOv3 network for training, and the vehicle object detection model is obtained. The vehicle object detection model can detect three types of vehicles: cars, buses, and trucks (Fig. 9). Because there are few motorcycles on the

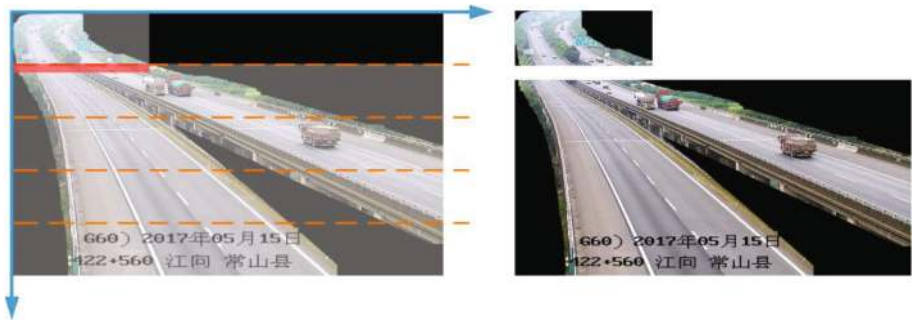


Fig. 8 Road surface segmentation

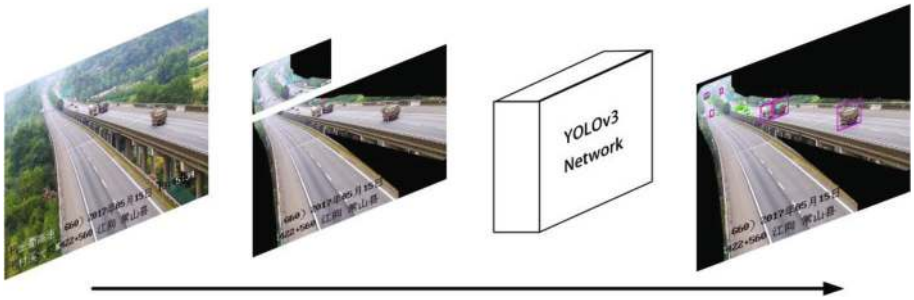


Fig. 9 Segmented image sent to the detection network and detected results merging. (Green, blue, and fuchsia boxes are labelled to indicate the "car", "bus", and "truck" regions, respectively.)

highway, they were not included in our detection. The remote area and proximal area of the road surface are sent to the network for detection. The detected vehicle box positions of the two areas are mapped back to the original image, and the correct object position is obtained in the original image. Using the vehicle object detection method for obtaining the category and location of the vehicle can provide necessary data for object tracking. The above information is sufficient for vehicle counting, and the vehicle detection method thus does not detect the specific characteristics of the vehicle or the condition of the vehicle.

Multi-object tracking

This section describes how multiple objects are tracked based on the object box detected in "Vehicle detection using YOLOv3" section. In this study, the ORB algorithm was used to extract the features of the detected vehicles, and good results were obtained. The ORB algorithm shows superior performance in terms of computational performance and matching costs. This algorithm is an excellent alternative to the SIFT and SURF image description algorithms. The ORB algorithm uses the Features From Accelerated Segment Test (FAST) to detect feature points and then uses the Harris operator to perform corner detection. After obtaining the feature points, the descriptor is calculated using the BRIEF algorithm. The coordinate system is established by taking the feature point as the centre of the circle and using the centroid of the point region as the x-axis of the coordinate system. Therefore, when the image is rotated, the coordinate system can be rotated according to the rotation of the

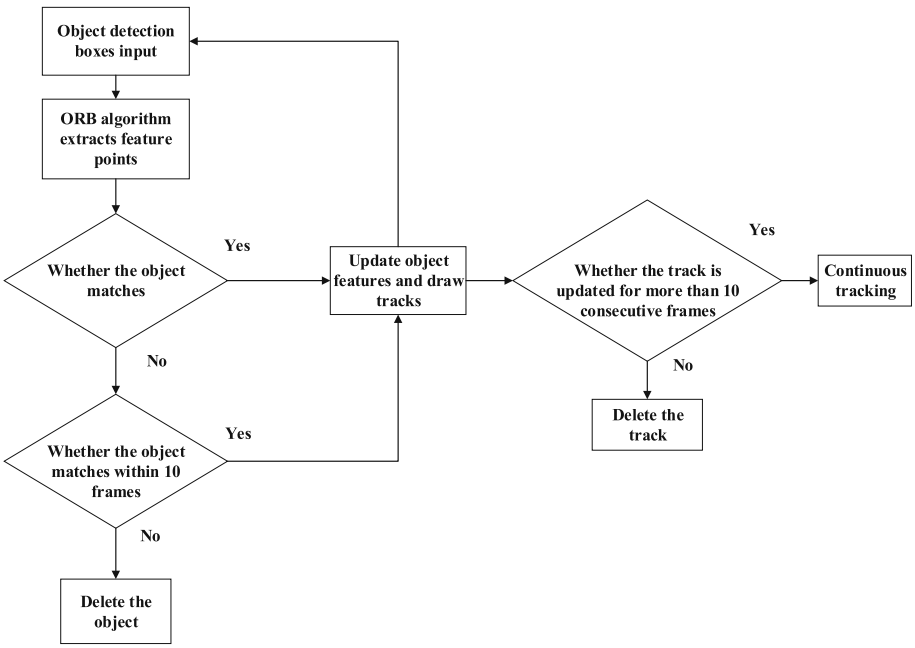


Fig. 10 Process of multi-object tracking

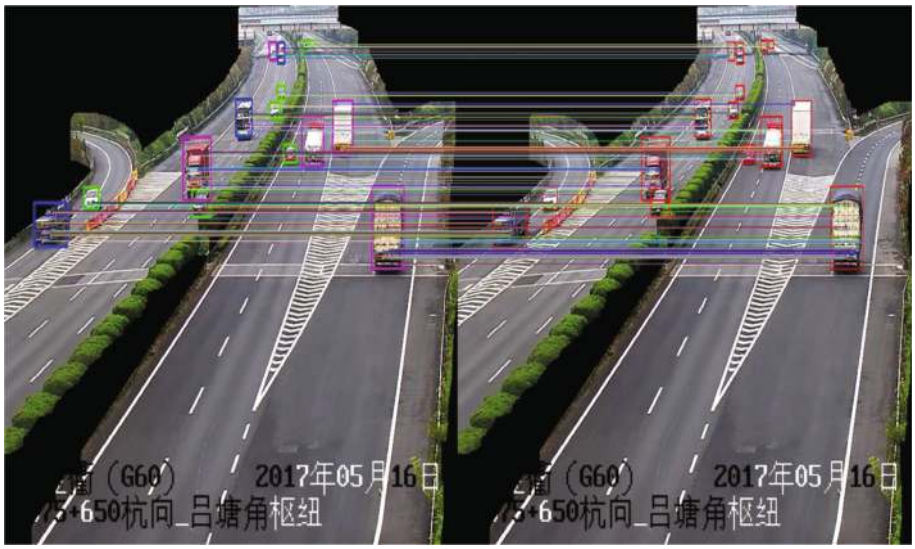


Fig. 11 Features of the detection object extracted by the ORB algorithm

image, and the feature point descriptor thus has rotation consistency. When the picture angle is changed, a consistent point can also be proposed. After obtaining the binary feature point descriptor, the XOR operation is used to match the feature points, which improves the matching efficiency.

The tracking process is shown in Fig. 10. When the number of matching points obtained is greater than the set threshold, the point is considered to be successfully matched and the matching box of the object is drawn. The source of the prediction box is as follows: feature point purification is performed using the RANSAC algorithm, which can exclude the incorrect noise points of the matching errors, and the homography matrix is estimated. According to the estimated homography matrix and the position of the original object detection box, a perspective transformation is performed to obtain a corresponding prediction box.

We used the ORB algorithm to extract feature points in the object detection box obtained by the vehicle detection algorithm. The object feature extraction is not performed from the entire road surface area, which dramatically

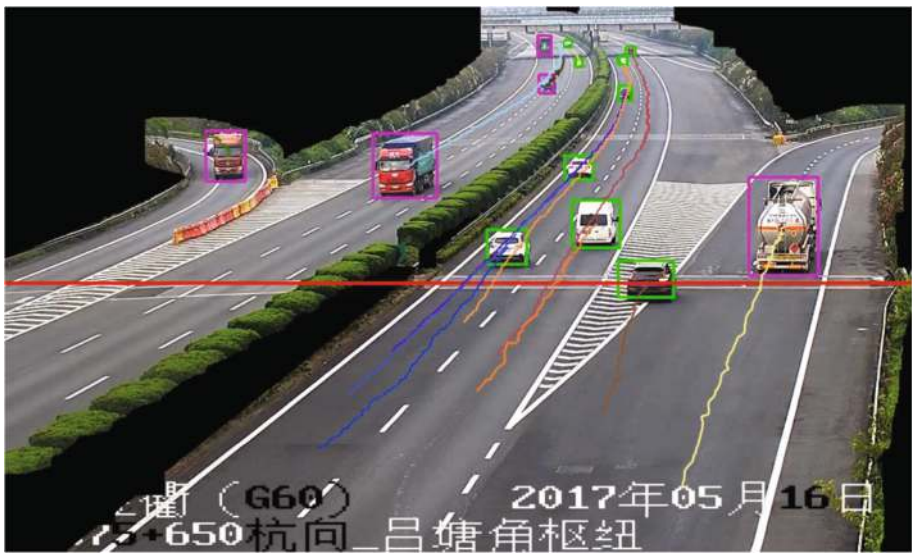


Fig. 12 Trajectory of the vehicle and detection line

reduces the amount of calculation. In object tracking, the prediction box of the object in the next frame is drawn since the change of the vehicle object in the continuous frame of the video is subtle according to the ORB feature extracted in the object box. If the prediction box and the detection box of the next frame meet the shortest distance requirement of the centre point, the same object successfully matches between the two frames (Fig. 11). We define a threshold T that refers to the maximum pixel distance of the detected centre point of the vehicle object box, which moves between two adjacent video frames. The positional movement of the same vehicle in the adjacent two frames is less than the threshold T. Therefore, when the centre point of the vehicle object box moves over T in the two adjacent frames, the cars in the two frames are not the same, and the data association fails. Considering the scale change during the movement of the vehicle, the value of the threshold T is related to the size of the vehicle object box. Different vehicle object boxes have different thresholds. This definition can meet the needs of vehicle movement and different input video sizes. T is calculated by Eq. 1, in which *box height* is the height of the vehicle object box.

$$T = \frac{box\ height}{0.25} \tag{1}$$

We delete the trajectory that is not updated for ten consecutive frames, which is suitable for the camera scene with a wide-angle of image collection on the highway under study. In this type of scene, the road surface captured by the camera is distant. In ten consecutive video frames, the vehicle will move farther away. Therefore, when the trajectory is not updated for ten frames, the trajectory is deleted. At the same time, the vehicle trajectory and the detection line will only intersect once, and the threshold setting thus does not affect the final counting result. If the prediction box fails to match in consecutive frames, the object is considered to be absent from the video scene, and the prediction box is deleted. From the above process, the global object detection results and tracking trajectories from the complete highway monitoring video perspective are obtained.

Trajectory analysis

This section describes the analysis of the trajectories of moving objects and the counting of multiple-object traffic information. Most of the highways are driven in two directions, and the roads are separated by isolation barriers. According to the direction of the vehicle tracking trajectory, we distinguish the direction of the vehicle in the world coordinate system and mark it as going to the camera (direction A) and driving away from the camera (direction B). A straight line is placed in the traffic scene image as a detection line for vehicle classification statistics. The detection line is placed at the 1/2 position on the high side of the traffic image (Fig. 12). The road traffic flow in both directions is simultaneously counted. When the trajectory of the object intersects the detection line, the information of the object is recorded. Finally, the number of objects in different directions and different categories in a certain period can be obtained.

Results and discussion

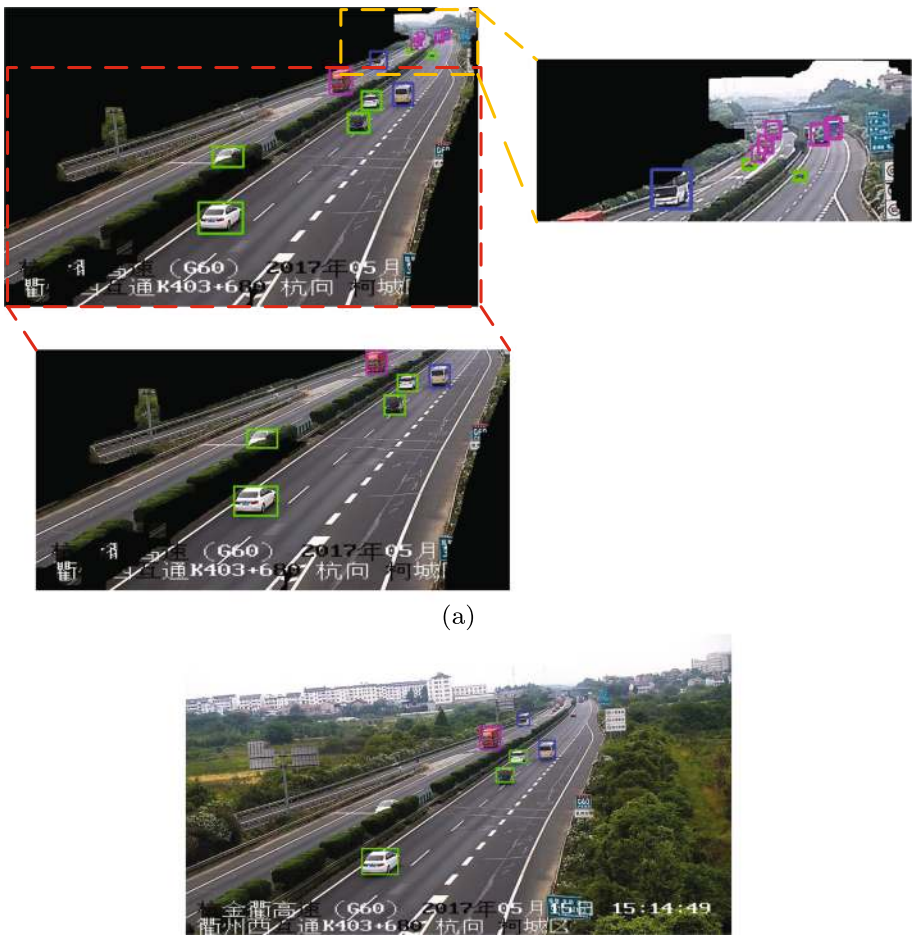
In this section, we describe the performance testing of the methods presented in "Methods" section. We experimented with the vehicle object dataset established in "Vehicle dataset" section. Our experiment used high definition highway videos for three different scenes, as shown in Fig. 1.

Table 2 Number of objects under different detection methods

Scenes	Video frames	Vehicle category	Total number of vehicle objects					
			Our method		Full-image detection method		Actual number of vehicles	
			Remote area	Proximal area	Remote area	Proximal area	Remote area	Proximal area
Scene 1	3,000	Car	6,128	8,430	493	6,616	6,849	8,550
		Bus	535	459	92	379	582	483
		Truck	5,311	5,320	840	4,703	5,792	5,471
Scene 2	3,000	Car	1,843	3,615	192	3,356	1,914	3,654
		Bus	194	364	82	295	207	382
		Truck	3,947	4,709	922	3,738	4,169	4,731
Scene 3	3,000	Car	1,774	2,336	224	2,188	1,834	2,352
		Bus	415	516	56	495	483	529
		Truck	3,678	3,490	731	2,662	3,726	3,507

Network training and vehicle detection

We used the YOLOv3 network for vehicle object detection and our established dataset for network training. In network training, there is no perfect solution for the dataset division. Our dataset dividing method follows the usual usage. We split the dataset into an 80% training set and a 20% test set. Our dataset has 11,129 images, the training set images, and the test set images are randomly selected from the dataset. Due to a large number of dataset pictures, the rate of the test set and training set is sufficient to obtain the model. To obtain an accurate model, the rate of the training set should be high. The training set has 8,904 images, and numerous vehicle samples can be trained to obtain accurate models for detecting cars, buses, and truck targets. The test set has 2225 images with vehicle targets that are completely different from the training set, which is sufficient to test the accuracy of the model that has been trained. We used a batch size of 32 and set the weight attenuation to 0.0005 and the momentum value to 0.9 for the maximum number of training iterations of 50,200. We used a learning rate of 0.01 for the first 20,000 iterations, which changed to 0.001 after 20,000 iterations. This approach made the gradient fall reasonably and made the loss value lower. To make the default anchor box more suitable for the dataset annotation box to be annotated, we used the k-means++ method to make changes. The training set of our dataset calculated the default anchor box size at the network resolution of 832*832, and we obtained nine sets of values: [13.2597, 21.4638], [24.1990, 40.4070], [39.4995, 63.8636], [61.4175, 96.3153], [86.6880, 137.2218], [99.3636, 189.9996], [125.6843, 260.8647], [179.7127 , 198.8155], [189.3695, 342.4765], with an average IOU of 71.20%. To improve the detection effect of small objects, we did not discard samples with less than 1-pixel value during training but put them into the network for training. We output the result of splicing the feature map of the previous layer of the routing layer before the last yolo layer of Darknet-53 and the 11th layer



(a)

(b)

Fig. 13 Single-frame video object detection results. Green, blue, and fuchsia boxes are labelled to indicate the "car", "bus", and "truck" regions, respectively. **a** Our method; **b** the full-image detection method

Table 3 Comparison of actual vehicle numbers by using different methods

	Vehicle category	Remote area		Proximal area		Average correct rate	
		Our method	Full-image detection method	Our method	Full-image detection method	Our method	Full-image detection method
Actual number of vehicles	Car	91.96%	8.58%	98.79%	83.54%	95.375%	46.06%
	Bus	89.94%	18.08%	96.05%	83.86%	92.995%	50.97%
	Truck	94.51%	18.21%	98.61%	80.99%	96.56%	49.6%
Overall correct rate		92.14%	14.96%	97.82%	82.80%	94.976%	48.86%

of Darknet-53. We set the step size to 4 in the upsampling layer before the last yolo layer. When we set the image input to the network, the network resolution was 832*832 instead of the default 416*416 resolution. After the input resolution is increased, when the network is output in the yolo layer, it can have a correspondingly larger resolution and can thus improve the accuracy of the object detection.

A continuous 3000 frames of images were used for vehicle detection in a variety of highway scenes by using our trained model. We extracted and divided the road surface area and put it into the network for vehicle detection. Then, we compared our method with the detection of images with 1920*1080 resolution into the network (without dividing the road surface); the results are shown in Table 2 and Fig. 13. We compared the number of object detections under different methods with the actual number of vehicles, as shown in Table 3.

Compared with the actual number of vehicles, our method comes close to the actual number of vehicles when the proximal area object of the road is large. When the object at the remote area of the road is small, the detection deviation is still less than 10%. The full-image detection method did not detect a large number of small objects in the remote area of the road. Our method effectively improves the detection of small objects in the remote area of the road. At the same time, in the proximal area of the road, our method is also better than the full-image detection method. However, the deviation is inaccurate. CNN may detect the wrong object or detect the non-object as an object, which results in an inaccurate total number of vehicles. Therefore, we calculated the average accuracy of the dataset in Table 4. Based on the 80% training set and 20% test set, we used the test set to calculate the model's average precision (*map*); *map* represents the average of the average accuracy (*ap*) of the total object *class number* (the *class number* in the experiment is 3). For each category, *ap* describes the average of 11 points for each possible threshold in the category's *precision*/ *recall* curve. We used a set of thresholds [0, 0.1, 0.2, ..., 1]. For *recall* greater than each threshold (the threshold in the experiment is 0.25), there will be a corresponding maximum precision $p_{max}(recall)$. The above 11 precisions are calculated, and *ap* is the average of these 11 $p_{max}(recall)$. We used this value to describe the quality of our model.

$$ap = \frac{1}{11}\sum_{recall=0}^{1} p_{max}(recall), \quad recall \in [0, 0.1, ..., 1],$$
$$map = \frac{\sum ap}{class\ number} \tag{2}$$

The calculation of *precision* and *recall* is as follows:

$$Precision = \frac{TP}{TP + FP},$$
$$Recall = \frac{TP}{TP + FN} \tag{3}$$

where TP, FN, and FP are the numbers of true positives, false negatives, and false positives, respectively. We obtained a final map value of 87.88%, which indicates that the method is a good way to locate and classify different vehicle objects. It can be concluded from the above analysis that the correct overall rate of our object detection is 83.46%, which indicates good location and classification of different vehicle objects and provides better detection results for multi-object tracking.

Tracking and counting

After obtaining the object box, we performed vehicle tracking based on the ORB feature point matching method and performed trajectory analysis. In the experiment, when the matching point of each object was greater than ten, the corresponding ORB prediction position was generated. Based on the direction in which the tracking trajectory was generated, we used the detection line to

Table 4 Accuracy of the network model

Parameters	ap			Precision	Recall	Average IoU	mAP
	Car	Bus	Truck				
Results	86.46%	88.57%	88.61%	0.88	0.89	71.32%	87.88%

judge the direction of motion of the vehicle and classify it for counting. We conducted experiments on the other three videos that are the same as the scene in "Network training and vehicle detection" section but with a different number of frames. We used the *real time rate* to evaluate the speed of the system proposed in this paper, which is defined as the ratio of the time required for the system to process a video to that of the original video played. In Eq. 4, the *system running time* is the time required for the system to process a video, and the *video running time* is the time required for the original video played. The smaller the *real time rate* value is, the faster the system performs the calculations. When the value of the *real time rate* is less than or equal to 1, the input video can be processed in real time.

$$real\ time\ rate = \frac{system\ running\ time}{video\ running\ time} \tag{4}$$

The results are shown in Table 5. The results show that the average accuracies of vehicle driving direction and vehicle counting are 92.3% and 93.2%, respectively. In the highway monitoring video, the car class has a small object and is easily blocked by large vehicles. At the same time, there will be multiple cars in parallel, which will affect the accuracy of the track counting. Our original video runs at 30 frames per second. From the calculation of the speed, it can be found that the vehicle tracking algorithm based on the ORB feature is fast. The system processing speed is related to the number of vehicles in the scene. The greater the number of vehicles, the more features need to be extracted, and the system processing time will thus become longer. In general, the vehicle counting system proposed in this manuscript is very close to real-time processing.

Conclusions

This study established a high-definition vehicle object dataset from the perspective of surveillance cameras and proposed an object detection and tracking method for highway surveillance video scenes. A more effective ROI area was obtained by the extraction of the road surface area of the highway. The YOLOv3 object detection algorithm obtained the end-to-end highway vehicle detection model based on the annotated highway vehicle object dataset. To address the problem of the small object detection and the multi-scale variation of the object, the road surface area was defined as a remote area and a proximal area. The two road areas of each frame were sequentially detected to obtain good vehicle detection results in the monitoring field. The position of the object in the image was predicted by the ORB feature extraction algorithm based on the object detection result. Then, the vehicle trajectory could be obtained by tracking the ORB features of multiple objects. Finally, the vehicle trajectories were analyzed to collect the data under the current highway traffic scene, such as driving direction, vehicle type, and vehicle number. The experimental results verified that the proposed vehicle detection and tracking method for highway surveillance video scenes has good performance and practicability. Compared with the traditional method of monitoring vehicle traffic by hardware, the method of this paper is low in cost and high in stability and does not require large-scale construction or installation work on existing monitoring equipment. According to the research reported in this paper, the surveillance camera can be further calibrated to obtain the internal and external parameters of the camera. The position information of the vehicle trajectory is thereby converted from the image coordinate system to

Table 5 Track counting results

Scenes		Scene 1			Scene 2			Scene 3			
Video frames		11000			22500			41000			Direction correct rate
Vehicle category		Car	Bus	Truck	Car	Bus	Truck	Car	Bus	Truck	
Direction A	Our method	29	21	3	110	40	21	287	141	22	0.92
	Actual number of vehicles	32	21	3	117	43	22	297	150	24	
	Extra Number	3	0	0	8	3	2	15	13	3	
	Missing number	0	0	0	1	0	1	5	4	1	
	Correct rate	0.906	1	1	0.923	0.930	0.864	0.933	0.887	0.833	
Direction B	Our method	41	37	4	117	69	13	300	168	15	0.931
	Actual number of vehicles	43	38	4	125	77	13	311	172	17	
	Extra Number	2	2	0	11	10	0	15	8	2	
	Missing number	0	1	0	3	2	0	4	4	0	
	Correct rate	0.953	0.947	1	0.888	0.844	1	0.939	0.930	0.882	
Real time rate			1.27			1.35			1.48		
Average correct rate			0.967			0.911			0.917		0.932

the world coordinate system. The vehicle speed can be calculated based on the calibration result of the camera. Combined with the presented vehicle detection and tracking methods, abnormal parking events and traffic jam events can be detected to obtain more abundant traffic information.

In summary, vehicles in Europe, such as in Germany, France, the United Kingdom, and the Netherlands, have similar characteristics to the vehicles in our vehicle dataset, and the angle and height of the road surveillance cameras installed in these countries can also clearly capture the long-distance road surface. Therefore, the methodology and results of the vehicle detection and counting system provided in this analysis will become important references for European transport studies.

Acknowledgments
Not applicable.

Authors' contributions
H-SS: Literature Search and Review, Manuscript Editing. H-XL: Content Planning and Analysis, Manuscript Writing. H-YL: Content Planning, Dataset Labeling. ZD: Literature Review, Manuscript Editing. XY: Data Collection, Dataset Labeling. All authors read and approved the final manuscript.

Funding
This work is supported by the National Natural Science Foundation of China (No.61572083),the Ministry of Education Joint Fund Project of China (No.614102022610),the Fundamental Research Funds for the Central Universities Team Cultivation Project(No.300102248402) and the Key Research and Development Program of Shannxi Province of China(No.2018ZDXM-GY-047).

Availability of data and materials
The vehicle dataset generated in Chapter 2 is available in the Google Drive, http://drive.google.com/open?id=1li858elZvUgss8rC_yDsb5bDfiRyhdrX. Other datasets analysed during the current study are not publicly available due privacy reasons. The datasets contain personal data that may not be publicly available. It was assured that data generated for the research project are only used for this research context.

Competing interests
On behalf of all authors, the corresponding author states that there are no competing interests.

Received: 2 July 2019 Accepted: 3 December 2019
Published online: 30 December 2019

References

1. Al-Smadi, M., Abdulrahim, K., Salam, R.A. (2016). Traffic surveillance: A review of vision based vehicle detection, recognition and tracking. *International Journal of Applied Engineering Research*, *11*(1), 713–726.
2. Radhakrishnan, M. (2013). Video object extraction by using background subtraction techniques for sports applications. *Digital Image Processing*, *5*(9), 91–97.
3. Qiu-Lin, L.I., & Jia-Feng, H.E. (2011). Vehicles detection based on three-frame-difference method and cross-entropy threshold method. *Computer Engineering*, *37*(4), 172–174.
4. Liu, Y., Yao, L., Shi, Q., Ding, J. (2014). Optical flow based urban road vehicle tracking, In *2013 Ninth International Conference on Computational Intelligence and Security*. https://doi.org/10.1109/cis.2013.89: IEEE.
5. Park, K., Lee, D., Park, Y. (2007). Video-based detection of street-parking violation, In *International Conference on Image Processing*. https://www.tib.eu/en/search/id/BLCP%3ACN066390870/Video-based-detection-ofstreet-parking-violation, vol. 1 (pp. 152–156). Las Vegas: IEEE.
6. Ferryman, J.M., Worrall, A.D., Sullivan, G.D., Baker, K.D. (1995). A generic deformable model for vehicle recognition, In *Procedings of the British Machine Vision Conference 1995*. https://doi.org/10.5244/c.9.13: British Machine Vision Association.
7. Han, D., Leotta, M.J., Cooper, D.B., Mundy, J.L. (2006). Vehicle class recognition from video-based on 3d curve probes, In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. https://doi.org/10.1109/vspets.2005.1570927: IEEE.
8. Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X. (2018). Object detection with deep learning: A review. arXiv e-prints, arXiv:1807.05511.
9. Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2014.81: IEEE.
10. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M. (2013). Selective search for object recognition. *International Journal of Computer Vision*, *104*(2), 154–171.
11. Kaiming, H., Xiangyu, Z., Shaoqing, R., Jian, S. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *37*(9), 1904–16.
12. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. (2016). Ssd: Single shot multibox detector, In *2016 European conference on computer vision*. https://doi.org/10.1007/978-3-319-46448-0_2 (pp. 21–37): Springer International Publishing.
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection, In *2016 IEEE conference on computer vision and pattern recognition*. https://doi.org/10.1109/cvpr.2016.91 (pp. 779–788): IEEE.
14. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D. (2014). Scalable object detection using deep neural networks, In *2014 IEEE conference on computer vision and pattern recognition*. https://doi.org/10.1109/cvpr.2014.276 (pp. 2147–2154): IEEE.
15. Redmon, J., & Farhadi, A. (2017). *Yolo9000: Better, faster, stronger*: IEEE. https://doi.org/10.1109/cvpr.2017.690.
16. Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
17. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection, In *2016 European conference on computer vision*. https://doi.org/10.1007/978-3-319-46493-0_22 (pp. 354–370): Springer International Publishing.
18. Hu, X., Xu, X., Xiao, Y., Hao, C., He, S., Jing, Q., Heng, P.A. (2018). Sinet: A scale-insensitive convolutional neural network for fast vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, *PP*(99), 1–10.
19. Palubinskas, G., Kurz, F., Reinartz, P. (2010). Model based traffic congestion detection in optical remote sensing imagery. *European Transport Research Review*, *2*(2), 85–92.
20. Nielsen, A.A. (2007). The regularized iteratively reweighted mad method for change detection in multi-and hyperspectral data. *IEEE Transactions on Image processing*, *16*(2), 463–478.
21. Rosenbaum, D., Kurz, F., Thomas, U., Suri, S., Reinartz, P. (2009). Towards automatic near real-time traffic monitoring with an airborne wide angle camera system. *European Transport Research Review*, *1*(1), 11–21.
22. Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *PAMI-8*(6), 679–698.
23. Asaidi, H., Aarab, A., Bellouki, M. (2014). Shadow elimination and vehicles classification approaches in traffic video surveillance context. *Journal of Visual Languages & Computing*, *25*(4), 333–345.
24. Negri, P., Clady, X., Hanif, S.M., Prevost, L. (2008). A cascade of boosted generative and discriminative classifiers for vehicle detection. *EURASIP Journal on Advances in Signal Processing*, *2008*, 136.
25. Fan, Q., Brown, L., Smith, J. (2016). A closer look at faster r-cnn for vehicle detection, In *2016 IEEE intelligent vehicles symposium (IV)*. https://doi.org/10.1109/ivs.2016.7535375: IEEE.
26. Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X., Kim, T.K. (2014). Multiple object tracking: A literature review. arXiv preprint arXiv:1409.7618.
27. Xing, J., Ai, H., Lao, S. (2009). Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses, In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvprw.2009.5206745 (pp. 1200–1207): IEEE.

28. Zhou, H., Yuan, Y., Shi, C. (2009). Object tracking using sift features and mean shift. *Computer Vision & Image Understanding*, *113*(3), 345–352.
29. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.R. (2011). Orb: an efficient alternative to sift or surf, In *2011 International Conference on Computer Vision*. https://doi.org/10.1109/iccv.2011.6126544: IEEE.
30. Luo, Z. (2018). Traffic analysis of low and ultra-low frame-rate videos, Doctoral dissertation. Université de Sherbrooke.
31. Geiger, A. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite, In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2012.6248074 (pp. 3354–3361): IEEE.
32. Zhe, Z., Liang, D., Zhang, S., Huang, X., Hu, S. (2016). Traffic-sign detection and classification in the wild, In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.232: IEEE.
33. Krause, J., Stark, M., Deng, J., Li, F.F. (2014). 3d object representations for fine-grained categorization, In *2013 IEEE International Conference on Computer Vision Workshops*. https://doi.org/10.1109/iccvw.2013.77: IEEE.
34. Yang, L., Ping, L., Chen, C.L., Tang, X. (2015). A large-scale car dataset for fine-grained categorization and verification, In *2015 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2015.7299023 (pp. 3973–3981): IEEE.
35. Zhen, D., Wu, Y., Pei, M., Jia, Y. (2015). Vehicle type classification using a semisupervised convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4), 2247–2256.
36. Guerrero-Gomez-Olmedo, R., Torre-Jimenez, B., Lopez-Sastre, R., Maldonado-Bascon, S., Ooro-Rubio, D. (2015). Extremely overlapping vehicle counting, In *Iberian Conference on Pattern Recognition & Image Analysis*. https://doi.org/10.1007/978-3-319-19390-8_48 (pp. 423–431): Springer International Publishing.

Publisher's Note