Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1969408, 16 pages
https://doi.org/10.1155/2020/1969408

Research Article

Video-Based Vehicle Counting for Expressway: A Novel Approach Based on Vehicle Detection and Correlation-Matched Tracking Using Image Data from PTZ Cameras

Qiao Meng,[1,2] Huansheng Song,[1] Yu'an Zhang,[2] Xiangqing Zhang,[1] Gang Li,[1] and Yanni Yang[1]

[1]*School of Information Engineering, Chang'an University, Xi'an, Shanxi 710064, China*
[2]*Computer Technology and Application Department, Qinghai University, Xining 810016, China*

Correspondence should be addressed to Qiao Meng; 250345481@qq.com

Received 29 September 2019; Revised 21 January 2020; Accepted 18 February 2020; Published 27 March 2020

Academic Editor: Volodymyr Ponomaryov

Vehicle counting plays a significant role in vehicle behavior analysis and traffic incident detection for established video surveillance systems on expressway. Since the existing sensor method and the traditional image processing method have the problems of difficulty in installation, high cost, and low precision, a novel vehicle counting method is proposed, which realizes efficient counting based on multivehicle detection and multivehicle tracking. For multivehicle detection tasks, a construction of the new expressway dataset consists of a large number of sample images with a high resolution (1920×1080) captured from real-world expressway scenes (including the diversity climatic conditions and visual angles) by Pan-Tilt-Zoom (PTZ) cameras, in which vehicle categories and annotation rules are defined. Moreover, a correlation-matched algorithm for multivehicle tracking is proposed, which solves the problem of occlusion and vehicle scale change in the tracking process. Due to the discontinuity and unsmooth of the trajectories that occurred during the tracking process, we designed a trajectory optimization algorithm based on least square method. Finally, a new vehicle counting method is designed based on the tracking results, in which the driving direction information of the vehicle is added in the counting process. The experimental results show that the proposed counting method in this research can achieve more than 93% accuracy and an average speed of 25 frames per second in expressway video sequence.

1. Introduction

As a core component of intelligent transportation, the real-time and effective vehicle counting method is of great significance for the expressway management department to implement traffic management and control violations. Since expressway has the characteristics of large traffic flow [1] and high speed, once traffic jam and parking events occur, it is easily to cause traffic accidents, which is extremely harmful to traffic safety. Over the past several years, video surveillance equipment was installed in all key sections of the expressways, but the vehicle counting is still based on sensors or just traditional image processing methods, which results in serious road damage, expensive information construction, and poor vehicle counting accuracy [2]. Therefore, it is of great theoretical and practical significance to make full use of existing monitoring resources and apply the methods of deep learning and computer vision to study the video-based vehicle counting method for traffic safety, traffic guidance, and traffic violation handling.

The widely applied methods of vehicle counting mainly include vehicle detection and vehicle tracking. The earliest vehicle detection was to extract the moving targets from the image sequence and identify the extracted targets. The vehicle detection method during this period mainly included background subtraction method [3, 4], frame difference method [5], and optical flow method [6]. However, the background subtraction method uses the weighted average method for background update, and the effect of background update will affect the integrity of the vehicle extraction and the accuracy of vehicle detection. Moreover, the frame difference method is greatly affected by the vehicle speed and the time interval of continuous frames. Furthermore, the optical flow method is a pixel-level density estimation, which is not suitable for real-time applications due to its large computation. In recent years, to solve the impact of complicated scenarios towards accurate target detection [7], machine learning methods and classification methods have been widely used before deep learning becomes the mainstream of computer vision. However, machine learning methods and classification methods have the disadvantages of high time complexity, weak region selection, and poor robustness of manually extracted features. Consequently, deep learning method is presented for target detection, which demonstrated that features extracted by the deep convolutional neural networks are more superior than hand-crafted features. Unlike machine learning, the existing target detection systems based on deep learning can be divided into proposal-based methods, such as R-CNN [8], SPP-net [9], Fast R-CNN [10], Faster R-CNN [11], and Mask R-CNN [12], and proposal-free methods, such as Single Shot Multibox Detector (SSD) [13] and You Only Look Once (YOLO) [14]. Unlike the proposal-based methods, SSD and YOLO use the method of setting the default box and the method of dividing the input image into a fixed grid to predict the target location and classification quickly, which makes the process of training and detecting much faster than the R-CNN series. However, SSD guarantees a fast detection speed, and its detection accuracy is far superior to YOLO. Moreover, the sufficient number of labelled training samples and the reasonable selection of representative training samples play a crucial role in SSD method [15]. On the whole, the comprehensive use of the effective deep learning models and dataset is of critical importance to improve the speed and accuracy of vehicle detection.

Vehicle tracking plays a significant role in vehicle counting and has attracted more and more attention in recent years. The existing approaches for vehicle counting based on videos can be categorized into deep learning-based tracking algorithms [16], online methods (MDP [17]), and batch-based methods (IOUT [18]). In practice, online methods and batch-based methods have difficulty in small target detection. Xiang et al. [17] applied online method to extract vehicles for target detection, but its measurement accuracy is seriously affected by occlusion if there are various types of vehicles and the distribution of vehicle speed is not uniform. Currently, the main methods for video-based vehicle tracking can be divided into generative method and discriminative method [19]. Sparse Coding was the mainstream of generative tracking [20] methods such as ALSA and L1APG [21] in previous years. Currently, as a representative of the discriminative tracking method, the correlation filtering method has gradually occupied the mainstream position and achieved satisfactory results, such as Kalman filter [22] and Kernel Correlation Filter (KCF) [23]. The existing deep learning-based tracking algorithms [16] are based on deep learning detection and then use KCF, Kalman filtering, and other algorithms for tracking. However, KCF and Kalman filtering need to infer the state of the current frame from the state of the previous frame, that is, to establish constraints by constructing a motion model, so as to obtain a group of possible candidate regions of target positions. This method is only suitable for single target tracking, but when multitarget tracking is performed, it is easy to produce tracking errors due to occlusion problems. Consequently, in order to solve the tracking difficulties caused by the variety of moving scenes, target occlusion, deformation, and vehicle scale changes, the design of an effective vehicle tracking algorithm plays an important role in improving the performance of the counting system.

This paper presents a new vehicle counting method based on expressway video sequence, which includes vehicle detection, vehicle tracking, and vehicle counting. Different from the natural images, a new dataset based on the expressway video sequence we designed is trained by the SSD model to effectively detect the various vehicles. Moreover, the proposed algorithm for the vehicle tracking can effectively deal with the trajectory problem caused by occlusion, deformation, and vehicle scale change. In order to solve the problem of trajectory breakage and nonsmoothing after tracking, a trajectory optimization algorithm is proposed, which is superior to the state-of-the-art methods. Finally, a new multivehicle counting algorithm with strong adaptability is designed. In summary, the contribution of this study includes the following points:

(1) A new dataset named NOHWY was constructed by the total number of 7849 RGB images in diverse climatic conditions captured by Pan-Tilt-Zoom (PTZ) cameras from expressway. For the sample annotation, LabelImg [24] tool was used to label the training images based on the annotation rules we defined.

(2) In order to solve the problem of trajectory point instability of moving vehicles, a novel vehicle correlation-matched algorithm is proposed for vehicle tracking, which considers the spatiotemporal distribution information of moving vehicles and effectively solves the problem of vehicle occlusion.

(3) A motion vehicle trajectory optimization algorithm based on least squares method is proposed, which effectively solves the problem of interruption and nonsmoothing problems in process of the vehicle tracking.

(4) A multivehicle counting method is designed, which realizes the counting of vehicles by the vehicle category in different driving directions.

This paper will be organized as follows. Section 2 introduces the construction process of the NOHWY dataset and the methodology for vehicle tracking and counting. Section 3 presents the experimental results and the analysis of the results and discussion. Section 4 makes a conclusion of the paper.

2. Materials and Methods

We present a novel video-based method (Figure 1) to count vehicles from expressway. Different from the traditional methods which only rely on the image processing to segment images and count vehicles, our approach is designed to achieve vehicle counting task through multitarget detection, multitarget tracking, and trajectory acquisition. Firstly, we collected RGB images in complex environments in expressways, including rainy weather, strong illumination, and insufficient night light. Meanwhile, a NOHWY dataset was constructed for SSD model training by labeling different types of vehicles. Furthermore, the original SSD model was adjusted from 300×300 to 512×512 for better detection and recognition. Secondly, based on the location information and scale variation characteristics of vehicle detection at different moments, a new vehicle correlation-matched algorithm is designed to track the motion trajectory of multiple vehicles. Moreover, due to problems such as interruption and nonsmoothing of the trajectory, the least squares method was employed to optimize the trajectory to obtain a good and accurate trajectory. Finally, we designed a new multivehicle counting algorithm, which can calculate the number of various types of vehicles, the current frame record, and the total number of vehicles in different driving directions.

2.1. Vehicle Detection

2.1.1. Dataset. Many studies have demonstrated that natural images can have good detection effects based on deep learning. Based on the advantages of the convolutional neural network (CNN), a variety of datasets such as ImageNet and PASCAL VOC [25] were designed to push the state-of-the-art target detection and classification results based on various networks. Different from these natural images, expressways have the characteristics of top-down view, known resolution, and linear motion. Hence, we conduct the following work to construct the NOHWY dataset.

(1) Data Collection. All images in this dataset are captured by the traffic monitoring software provided by Hangzhou Huijing Technology Co., Ltd., which realizes the online monitoring of the expressway through the Pan-Tilt-Zoom (PTZ) camera installed on the Hangzhou expressway. Actually, the installation height of PTZ cameras on the expressway is about 12 meters. Moreover, the camera's focal length, pitch angle, and deflection angle are variable; that is, the shooting angle can rotate freely without preset position. Since the traffic monitoring software provides an image acquisition interface for online sample collection, as shown in Figure 2, it can remotely monitor the expressway sections corresponding to 25 PTZ cameras within 24 hours. According to the working principle of the software, we need to set the interface number corresponding to a PTZ camera in advance, after the monitoring is turned on, the image can only be saved at any time through manual selection. Hence, a NOHWY dataset we designed is a collection of images with a resolution of 1920×1080 captured by the traffic monitoring software under different climate condition, different scenes, and different angles. In addition, we select samples according to the principle of sample balance to solve the problem of more cars and fewer other types of vehicles in expressway scenarios. However, in the case of large similarity between samples, it is necessary to delete the redundant samples. Consequently, the final sample set needs to satisfy the characteristics of rich scenes, diverse angles, and complex climatic environment.

(2) Vehicle Annotation. Referring to the domestic automobile standard classification manual, vehicles on expressway can be divided into three categories: truck, bus, and car. Moreover, there are six PhD students using LabelImg annotation tool to label our dataset for three months, which can effectively reduce the annotation error rate. However, the strengths and weaknesses of the dataset rely on the definition of annotation rules, and annotation rules are of great significance to network training. Based on the characteristics of expressway traffic, we appoint annotation rules as follows:

Small target: the farther away from the camera, the less characteristics of the vehicle, so the small targets in the distance are not labelled (Figure 3(a)). In our approach, vehicles that appear in one-third of the distant scenes are defined as small targets that do not need to be labelled.

Target occlusion: the objective of the rule is to set occlusion rate in each image to ensure the effective extraction of multi-scale spatial features. If the occluded area of the vehicle is more than 1/2 of its own area, the vehicle is not labelled (Figure 3(b)).

Target truncation: target truncation means that the target is truncated by the boundaries of the image. Therefore, if the vehicle area outside the image is more than 2/3 of its own area, the vehicle is not labelled (Figure 3(c)).

Special samples: a few samples have the problem of category ambiguity. The samples can be labelled as difficult (difficult samples) and the field is set to 1, which means that this is an object rather than multiple objects. As shown in Figure 3(d), object 1 with red sign is labelled as one vehicle instead of multiple vehicles, so it can be detected and classified as a vehicle during training.

In NOHWY dataset, we employed the real-time online acquisition platform of Figure 2 to capture samples. In Figure 2, the "channel" button corresponds to 25 PTZ camera channels. You can select a certain camera by setting the "channel" value. The installation positions of these 25 PTZ cameras include top-view, side-view, and front-view. Since PTZ cameras can rotate and adjust its own angle according to the position information of the moving target, therefore, we used the letters A to E to represent 5 angles of rotation by the PTZ camera. Among them, A, B, C, D, and E

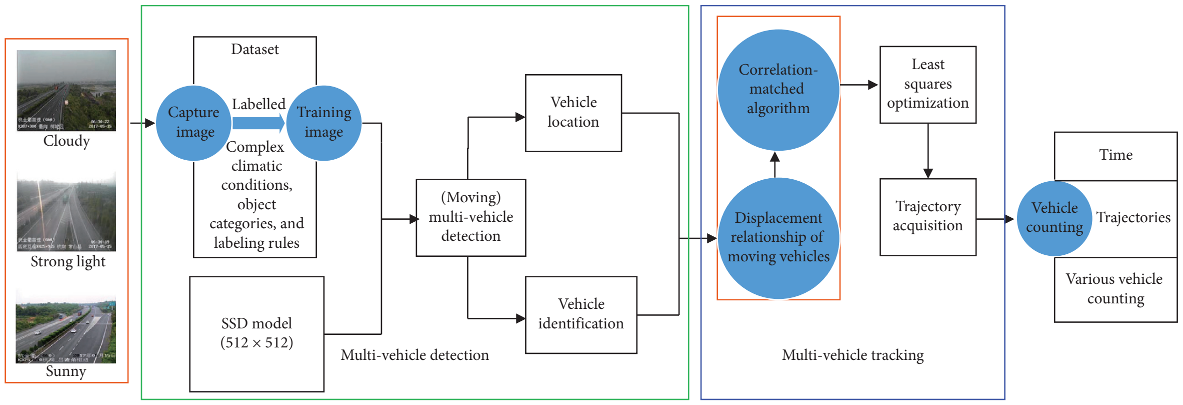
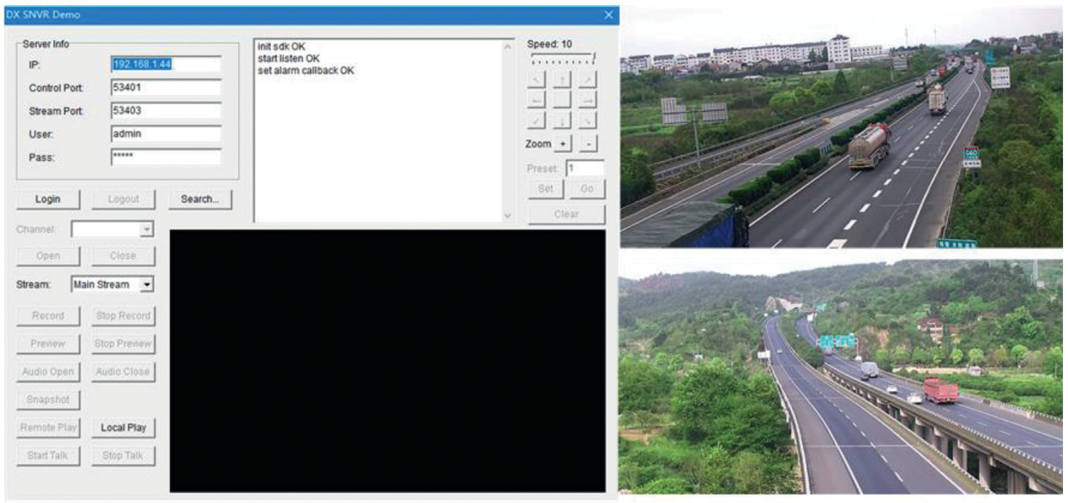


Figure 1: Framework of our proposed approach. It contains dataset construction, moving multivehicle detection, multivehicle tracking, trajectory acquisition, and vehicle counting.



IP: 192.168.1.44
Control Port: 53401
Stream Port: 53403
User: admin
Pass: *****
Stream: Main Stream
Speed: 10
Preset: 1

init sdk OK
start listen OK
set alarm callback OK

Figure 2: Online image acquisition interface of traffic monitoring software. The figure on the left is the image acquisition interface, which can use the "open" and "snapshot" buttons to obtain the expressway image at any time. The figure on the right shows examples of the training images collected by the traffic monitoring software.

represent the rotation of PTZ camera to 30°, 45°, 60°, 90°, and 120°, respectively. Therefore, we employed five doctoral students to collect a total of 7849 samples with a resolution of 1920×1080 using the online acquisition platform of Figure 2, and each doctoral student was responsible for five camera channels. We randomly selected 1962 samples as the test set and the rest 5887 samples as the training set labelled by the annotation tool of LabelImg. The statistical results of the number of classified samples in different scenarios are presented in Table 1.

2.1.2. *SSD Model.*

SSD is a high accuracy and fast speed target detection algorithm, which is proposed for real-time detection. SSD is built on a basic network that ends with some convolutional layers. SSD predicts the category scores and location offsets for the default bounding boxes by two 3×3 convolutional layers. In order to detect targets at different scales, SSD adds a series of progressively smaller convolutional layers to construct pyramid feature maps and sets aspect ratios for adjusting varying target shapes according to the receptive field size of the layers. Moreover, non-maximum suppression (NMS) is used to post-process the final predictions to get the detection results. In SSD model, SSD detector uses VGG [26] as the base network, and each input image needs to be resized to a fixed size (300×300). However, most of the images on the expressway are high-definition images. If the image is directly resized to 300×300, the image will be blurred. Therefore, we modify the 300×300 model to 512×512 (Figure 4). After analysis, our 512×512 SSD model can achieve real-time vehicle detection, and its processing speed is faster than other most advanced target detectors.

2.2. *Vehicle Tracking*

2.2.1. *Correlation-Matched Algorithm for Vehicle Tracking.*

Vehicle detection is a requisite process for vehicle tracking. Since existing tracking algorithms such as KCF can only track one type of vehicle, we design a new multi-vehicle tracking algorithm based on the correlation-matched

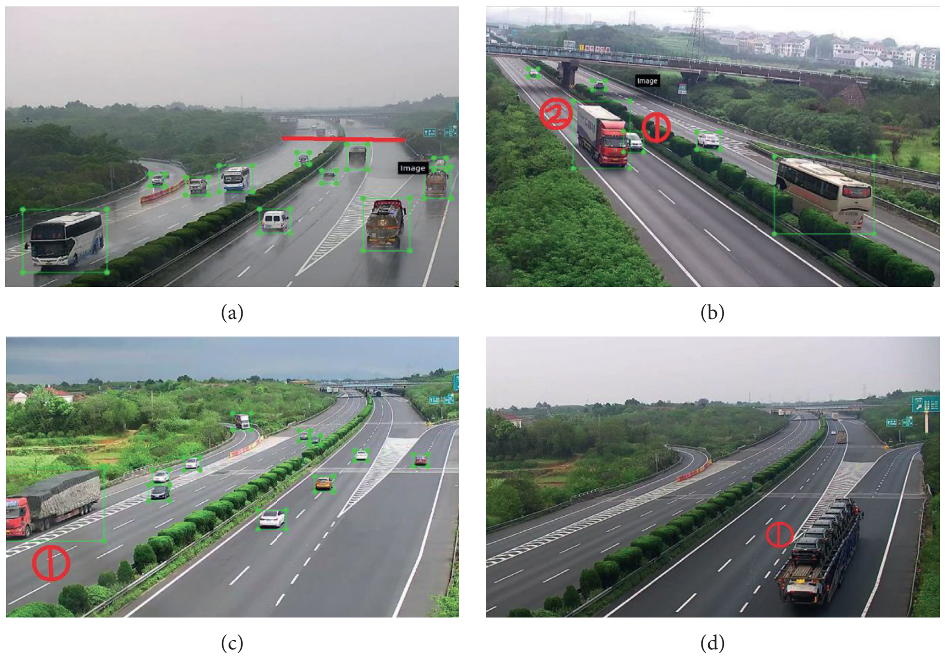


Figure 3: Examples showing our proposed annotation rules. (a) The red line in the figure represents the demarcation line of the labelled area, and the small vehicle located outside the red line is not labelled. (b) Truck (the red signs 2) occludes car (the red signs 1) less than 1/2 of car's own area, so car is labelled. (c) Truck (the red signs 1) is truncated less than 2/3 of its own area, so the truck is labelled. (d) The description of a special sample (the red signs 1) which is labelled as difficult.

Table 1: Classification statistics in different scenarios.

Annotation	*A* Sunny	*B* Cloudy	*C* Rainy	*D* Foggy	*E* Night
Car	511	324	223	313	287
Bus	398	671	423	361	119
Truck	423	316	588	348	582

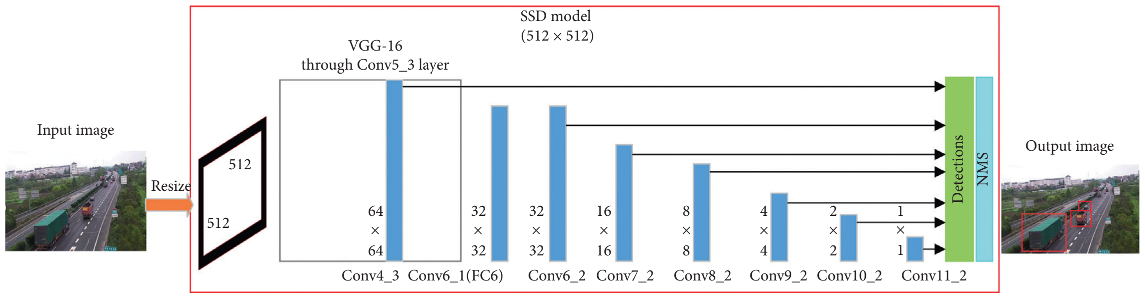


Figure 4: Framework of SSD model (512 × 512) [13]. As is shown, the red box in the figure contains the structure of our SSD 512 × 512 model. If an image is input, it will be resized to 512 × 512. Moreover, VGG-16 is still employed as the base network for the SSD model to extract features.

method of multi-vehicle detection information. In the previous section, the SSD model effectively learns the characteristics of vehicles in expressway scenes by training the NOHWY dataset, and it can efficiently provide accurate detection results for vehicle tracking, including vehicle category information and vehicle location information.

Since the targets detection in the previous chapter provided category information and four-parameter information of the target bounding box for each detected target, the four parameters are the width and height of the target bounding box and the center point coordinate values are expressed as (x, y, w, h). The study found that the Euclidean distance between two adjacent frames of the same vehicle is the smallest when it is driving on the expressway. Therefore, the main idea of vehicle tracking algorithm is derived based on video sequence from expressway. We represent the vehicle bounding boxes detected by the SSD model as a four-dimensional vector $V_t(x_t, y_t, w_t, h_t)$, where t denotes a vehicle, $t = (1, \ldots, i, j, \ldots, n)$, x and y represent the coordinates of the center point of the bounding box, and w and h represent the width and height of the bounding box, respectively. Assuming that the ith frame is adjacent to the jth frame, we need to calculate the minimum Euclidean distance between each bounding box detected in the ith frame and all the bounding boxes detected in the jth frame. The calculation formula is as follows:

$$p_{i1} = \min \cdot \left(\sqrt{\left(x_{i1} - x_{j1}\right)^2 + \left(y_{i1} - y_{j1}\right)^2}, \ldots, \sqrt{\left(x_{i1} - x_{jk}\right)^2 + \left(y_{i1} - y_{jk}\right)^2} \right), \tag{1}$$

where i and j represent the ith frame and the jth frame, respectively, and k represents the last bounding box in the jth frame. Similarly, the minimum Euclidean distance between each bounding box in the ith frame and all bounding boxes in the jth frame is calculated. In addition, we connect the center point of the two bounding boxes in adjacent frames, which need to satisfy the condition that the Euclidean distance of the bounding box center point in the adjacent frame is the smallest. However, it is not possible to obtain the correct trajectory of each vehicle by applying this method alone. Because the expressways are not single-lane roads, if the vehicle occlusion problem occurs, although the minimum distance between adjacent frames is satisfied, it will cause the track connection to be incorrect.

Through our research, the solution to solve this problem is to add another important constraint condition to judge the trajectory points. Generally, the bounding box of the same vehicle has the largest overlap area between adjacent frames. Therefore, we calculate the overlap area U of the bounding box between adjacent frames. The larger the overlap area of the two bounding boxes, the larger the U value. The U value between the ith frame and the jth frame is calculated as follows:

$$U_{i1} = \max\left(\left(w_{i1} * h_{i1}\right) \cap \left(w_{j1} * h_{j1}\right), \ldots, \left(w_{i1} * h_{i1}\right) \cap \left(w_{jk} * h_{jk}\right)\right), \tag{2}$$

This formula calculates the maximum value of the overlap area between each bounding box detected in the ith frame and all the bounding boxes detected in the jth frame. Considering the constraints of formula 1 and formula 2, we stipulate that the connection of the central point of the bounding box will be performed only when formula 1 and formula 2 are satisfied at the same time.

Through the above approach, we can achieve vehicle tracking perfectly, especially when large vehicles occlude the small vehicles. However, when the same type of vehicles occludes each other or runs in parallel, the algorithm will have an error (Figure 5). To solve this problem, we need to add the driving direction information of the vehicle and set the angle threshold to enhance the judgment of the trajectory points.

We define the set of trajectory points satisfying the tracking conditions of formula 1 and formula 2 as $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S_n}\}$. Since the detection efficiency of the SSD model is 28 frames per second (FPS), the same vehicle can obtain the position coordinates of 28 bounding box center points per second. According to the characteristics of the expressway, the vehicle can be considered as a linear motion, and the calculation of the angle between adjacent trajectory points is added to the existing tracking algorithm. The smaller the angle value, the greater the possibility that same vehicle will occlude each other (Figure 5(c)). Based on this theory, the adjacent trajectory points x, y, and z are connected, and the angle between two straight lines connected by three adjacent trajectory points is calculated. Moreover, we set a threshold for the angle to exclude abnormal trajectory points. The calculation formula is as follows:

$$\boldsymbol{\alpha_t} = \mathbf{arccos}\left(\frac{\left|S_t - S_{t-1}\right|^2 + \left|S_{t+1} - S_t\right|^2 - \left|S_{t+1} - S_{t-1}\right|^2}{2\left|S_t - S_{t-1}\right|\left|S_{t+1} - S_t\right|} \right), \tag{3}$$

$$y = \rho \times \frac{\sum_{t=2}^{n-1} \alpha_t}{n-2}, \quad \rho \in [0.5, 0.7], \tag{4}$$

where α_t represents the angle between adjacent trajectory points. S_{t-1}, S_t, and S_{t+1} represent three adjacent trajectory points. y is a threshold, ρ represents the weight, and n represents the number of the trajectory points. When the angles (Θ and γ in Figure 5(c)) produced by the abnormal trajectory points (red points in Figure 5(c)) are relatively small, the range of ρ is set to $[0.5, 0.7]$. That is to say, if a targets' trajectory points number is n, then calculate the angle value of each adjacent three trajectory points and find the average value of all the angle values. In addition, the final threshold is equal to the result of the average of the angle times the weight, where the range of weights is defined as $[0.5, 0.7]$. As the vehicle is moving in a straight line, the angle value of the abnormal trajectory points caused by the same type of vehicle running in parallel will not exceed 0.5 times of the average value. Therefore, the weight in this article is 0.5. Thus, if the value of α_t is equal to 0, it means that a congestion or parking event occurs. In addition, if the value of α_t is between 0 and y, it is considered that the trajectory point is an abnormal trajectory point, which needs to be deleted and the judgment of tracking conditions should be continued.

2.2.2. Trajectory Optimization by Least Squares Method. Since the camera will jitter on the expressway, the trajectory between adjacent frames is often not smooth. Generally, the motion of the vehicle on the expressways has a characteristic of local linearity. Therefore, the least square method is used to fit the trajectory points to solve the problem of trajectory smoothness. Due to reducing the trajectory deviation, our trajectory optimization algorithm based on least squares is performed in a piecewise linear fashion, and every ten

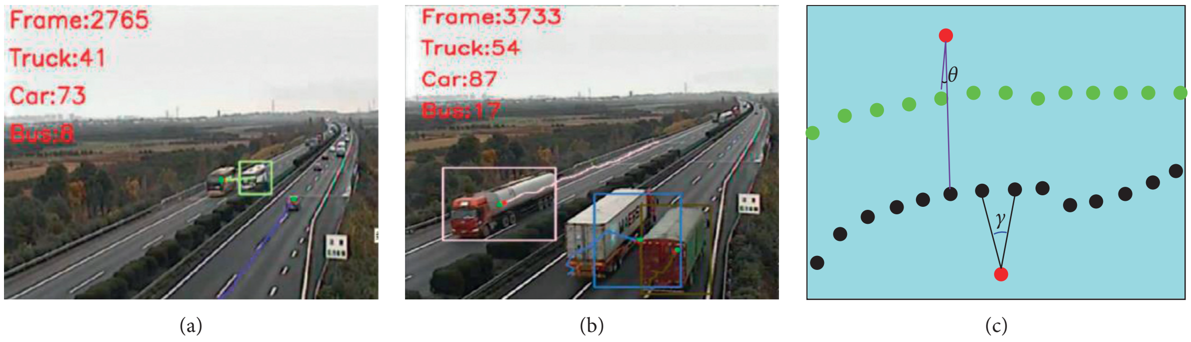


FIGURE 5: Example errors about our tracking algorithms and schematic diagram of abnormal trajectory points. We define that the forward direction is from near to far and the backward direction is from far to near. (a) The two buses occlude in the backward direction, satisfying the constraints of the tracking algorithm, but the trajectory connection is error. (b) The two trucks travelling in parallel in the forward direction also satisfy the constraints of the tracking algorithm so that the trajectory connection is wrong. (c) A schematic representation of the trajectory points of two trucks travelling in parallel for a period of time.

trajectory points were performed by the optimization algorithm.

A series of trajectory points $S = \{S_1, S_2, \ldots, S_k, \ldots, S_n\}$ are obtained by using the correlation-matched algorithm, where S_k is the central point of the detection bounding box and its coordinates are expressed as (m_k, v_k). Suppose that the linear fitting equation is as follows:

$$p = f(x) = qx + c, \tag{5}$$

and when the deviation between the value of v_k and $f(x_k)$ on the straight line is minimum, the fitting formula is optimal. Based on this theory, the values of q and c in the linear equation are solved:

$$\sum_{k=1}^{n} (v_k - f(x_k))^2 = \sum_{k=1}^{n} (v_k - (qx_k + c))^2 = 0. \tag{6}$$

Solve equations as follows:

$$\begin{cases} -2\sum_{k=1}^{n} (v_k - qm_k - c) = 0, \\ -2\sum_{k=1}^{n} (v_k - qm_k - c)m_k = 0. \end{cases} \tag{7}$$

$$\begin{cases} q = \dfrac{\sum_{k=1}^{n} m_k \sum_{k=1}^{n} v_k - n\sum_{k=1}^{n} m_k v_k}{\left(\sum_{k=1}^{n} m_k\right)^2 - n\sum_{k=1}^{n} m_k^2}, \\ c = \dfrac{\sum_{k=1}^{n} m_k v_k \sum_{k=1}^{n} m_k - \sum_{k=1}^{n} v_k \sum_{k=1}^{n} m_k^2}{\left(\sum_{k=1}^{n} m_k\right)^2 - n\sum_{k=1}^{n} m_k^2}. \end{cases} \tag{8}$$

The fitted straight line is obtained by taking the values of q and c into the straight line equation. Finally, the trajectory is smoothed by least square fitting (Figure 6).

2.3. Vehicle Counting. Considering the original approach for vehicle counting is based on virtual test lines [27], which is to set up virtual test lines on the road, we design an algorithm that can be applied to multi-vehicle counting, which simultaneously counts different categories of vehicles.

Based on the local linearity of the expressway, we set up a vehicle counting area according to the shooting range of the camera (Figure 7(a)). The driving direction of vehicles can be divided into the forward direction and the backward direction. Since the multi-vehicle tracking algorithm we designed, it is not necessary to design multiple detection lines to determine whether there is a vehicle passing through. We divide the counting area into four regions, A, B, C, and D, and establish a coordinate system for the counting area (Figure 7(b)). Moreover, the center of the counting region is set to the origin of the coordinate system, and the four regions A, B, C, and D correspond to the four quadrants of the coordinate system, respectively. Suppose that the coordinates (coordinates of the center points of the detection bounding box) of the starting and ending points of the vehicle trajectory points in the counting area are expressed as $S(x, y)$ and $S'(x', y')$. It is necessary to calculate the angle between the start point of the trajectory and the x-axis and the angle between the end point of the trajectory and the x-axis, respectively. The calculation formula is as follows:

$$\begin{cases} \tan\theta_1 = \dfrac{-y}{x} \\ \tan\theta_2 = \dfrac{y}{x} \\ \tan\theta_3 = \dfrac{y}{-x} \\ \tan\theta_4 = \dfrac{-y}{-x} \end{cases} \xrightarrow{\text{solution}} \begin{matrix} \theta_1 = -\arctan\left|\dfrac{-y}{x}\right| \\ \theta_2 = \arctan\left|\dfrac{y}{x}\right| \\ \theta_3 = -\arctan\left|\dfrac{y}{-x}\right| \\ \theta_4 = \arctan\left|\dfrac{-y}{-x}\right| \end{matrix}. \tag{9}$$

If the angle satisfies the condition that θ_1 is a negative value and θ_2 is a positive value, then the forward counter adds 1. Similarly, if θ_3 is a negative value and θ_4 is a positive value, the backward counter adds 1.

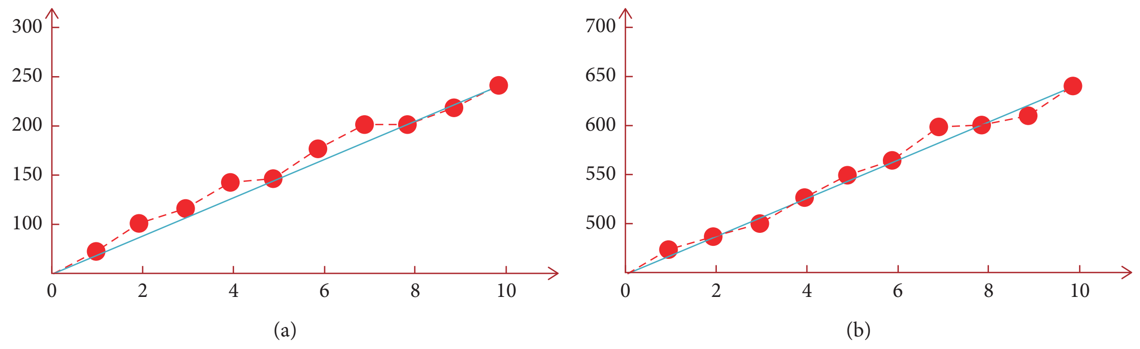


Figure 6: Examples of smoothing results for trajectory points. (a) The abscissa indicates the number of trajectory points, and the ordinate indicates the abscissa of the center point of bounding boxes. (b) The abscissa indicates the number of trajectory points, and the ordinate indicates the ordinate of the center point of bounding boxes.

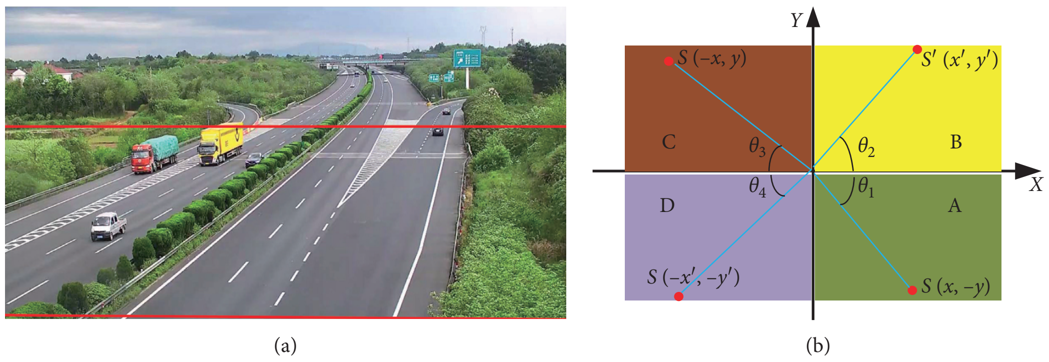


Figure 7: Examples of multivehicle counting methods. (a) Set the counting area. The area surrounded by the red line is defined as the count area. (b) A graph of the counting method. Different colors represent different regions, such as region A representing the fourth quadrant, region B representing the first quadrant, region C representing the second quadrant, and region D representing the fourth quadrant. The trajectory points of different regions satisfy the coordinate characteristics of the corresponding quadrants.

3. Results and Discussion

3.1. Experimental Results and Evaluation of Vehicle Detection. To evaluate the effectiveness of the vehicle detection method we proposed, we applied the method aforementioned in Section 2.1 to the Hangzhou expressway for training and testing. The vehicle detection method includes two key issues, one is the dataset, and the other is the deep learning model. In Section 2.1, by analyzing the necessity of constructing an expressway dataset, we define the annotation rules for the dataset named NOHWY. Moreover, we adjusted the parameters of SSD model during the training process to suit the environment of the expressway.

3.1.1. Training

(1) Preparation of Basic Work. The machine we used with CPU Inter Core i9-9900X 10 Core/3.50 GHz/19.25 MB and NVIDIA RTX 2080TI GPU named DEVMAX402. Our experiments were based on Caffe framework, which is extensively used in deep learning. Moreover, NOHWY dataset was prepared for training, including 5887 training samples with a resolution of 1920×1080 and their label files. After analyzing the size of vehicles (trucks, buses, and cars) on the collected images, we choose the SSD 512×512 model as shown in Figure 4 for training and testing.

(2) Data Augmentation. For the training processing of SSD model, besides the original image, image clipping, brightness adjustment, and noise addition are used to data augmentation. For example, the image clipping in Figure 8 is to better detect the small targets in large images, while adjusting the brightness of the image is conducive to the detection of night scene. In addition, adding noise to the training samples can improve the robustness of the SSD detector.

(3) Setting Aspect Ratios of the Default Boxes. Considering the characteristics of trucks, cars, and buses, we set six types of aspect ratios (1/1, 3/1, 4/1, 1/3, and 1/4) for the default box to make the SSD model fit better to trucks and buses with a shape other than square. During the process of the training, since each ground truth box can match different default boxes, we must set an overlap value to indicate the largest IoU. Therefore, when the overlap value is over 0.6, the default boxes can be considered as “matched”.

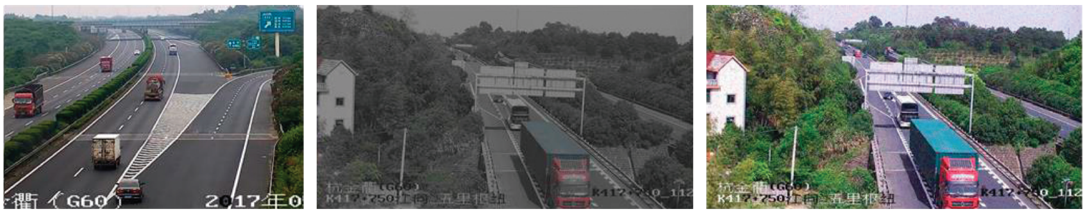


Figure 8: Examples of data augmentation. Left: image clipping. Middle: adjust image brightness Right: add Gaussian noise.

(4) Setting Training Parameters. Since the number of iterations is too small, the convergence cannot be achieved. Conversely, if the number of iterations is too large, overfitting occurs. Therefore, after repeated verification, the number of iterations is set to 120,000, and the number of Epochs is 652. However, the learning rate is an important parameter for the training of deep learning models; it directly affects the quality of model training. Many experiments have proved that using the same learning rate all the time will lead to training which cannot converge, and the detection accuracy of the model declines. Therefore, this paper uses the piecewise method to set the learning rate; that is, in the first 60,000 iterations, the value of learning rate is set to 0.001, and in the last 60,000 iterations, the value of learning rate is set to 0.001×0.05.

3.1.2. Testing and Results. In the vehicle detection method, based on our parameter setting method, the SSD 512×512 model is applied to train NOHWY dataset, and the obtained training model preserves all the parameters generated during the training process. In order to validate our training model, beside some of the test set, we use 15 video sequences to test, which are selected from over 30 minutes taken by the traffic monitoring software provided by Hangzhou Huijing Technology Co., Ltd. The video with the resolution of 1920×1080 including a variety of different climatic conditions. Figure 9 shows our training and testing results on NOHWY dataset.

As the criteria, mean average precision (mAP) is the average of three categories, Aps, recall, and accuracy ware used to evaluate expressways' vehicle detection. In our experiment, recall and precision curve were drawn for each vehicle, in which Average Precision (AP) measures the quality of SSD 512×512 model in each category. Moreover, we draw curves based on test accuracy, training time, and training loss for training and testing results. Moreover, some parameters are used to calculate the criteria mentioned before. Therefore, AP and accuracy can be calculated as follows:

$$\begin{aligned} \mathbf{AP} &= \frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FN}}, \\ \mathbf{Accuracy} &= \frac{\mathbf{TP}+\mathbf{TN}}{\mathbf{TP}+\mathbf{TN}+\mathbf{FP}+\mathbf{FN}}. \end{aligned} \tag{10}$$

TN means true negatives, which represents the number of negative samples predicted correctly. TP means true positives, which represents the number of positive samples predicted correctly. FN means false negatives, which represents the number of negative samples predicted incorrectly. FP means false positives, which represents the number of positive samples predicted incorrectly.

As shown in Figure 9, it has a test accuracy of more than 93% on the test experiment in Figure 9(a), which is much higher than the value of 76.88% from SSD 300×300 model. The loss in Figure 9(b) is lower than 1, which guarantees the performance of the model. In the test process, the GPU which is used from NVIDIA 1080Ti can predict time of SSD 512×512 which is 28 frames per second (FPS). Moreover, we used AP to describe the vehicle detection models' performance. In Figure 9(c), our dataset trained by SSD 512×512 model has a higher mAP value, and the recalls are also more than 90% on the experiment. Compared with SSD 300×300 model, the modified SSD 512×512 model not only eliminates the system crash caused by too many training parameters but also improves the information loss. Since the NOHWY dataset has a good adaptability to the expressway, the improved SSD 512×512 model not only achieves good detection results but also guarantees the speed of detection and meets the real-time requirements of expressways.

In order to further illustrate that the SSD 512×512 model also guarantees the detection effect under the condition of ensuring real-time requirements, we compare it with the Faster-RCNN method with high detection accuracy and the YOLO method with fast detection speed. As shown in Figure 9(d), the SSD 512×512 model has performed well on our dataset.

3.1.3. Evaluation. To further demonstrate the robustness of the NOHWY dataset, we conduct more experiments to compare the adaptability of our dataset to the state-of-the-art datasets. Therefore, we used the mainstream dataset or the latest dataset currently applied to vehicle detection tasks to compare our dataset. We choose KITTI [28], UAV-DT [29], and UA-DETRAC [30] datasets for experiments. Among them, KITTI dataset is currently the largest international dataset for computer vision algorithm evaluation in autonomous driving scenarios. UAV-DT dataset is a complex scene dataset for Unmanned Aerial Vehicle (UAV) recognition and tracking tasks. UA-DETRAC dataset is a challenging real-world multi-target detection dataset. In Table 2, we compared the three datasets with NOHWY dataset in detail. As shown in Table 2, image represents the total number of images in the dataset, and year represents the time the dataset was published.

To further compare the adaptability of our dataset, we used the SSD 512×512 model to train the dataset of KITTI, UAV-DT, and UA-DETRAC and prepared six expressway videos over one hour in different time periods to test the training model. It should be noted that we only selected the

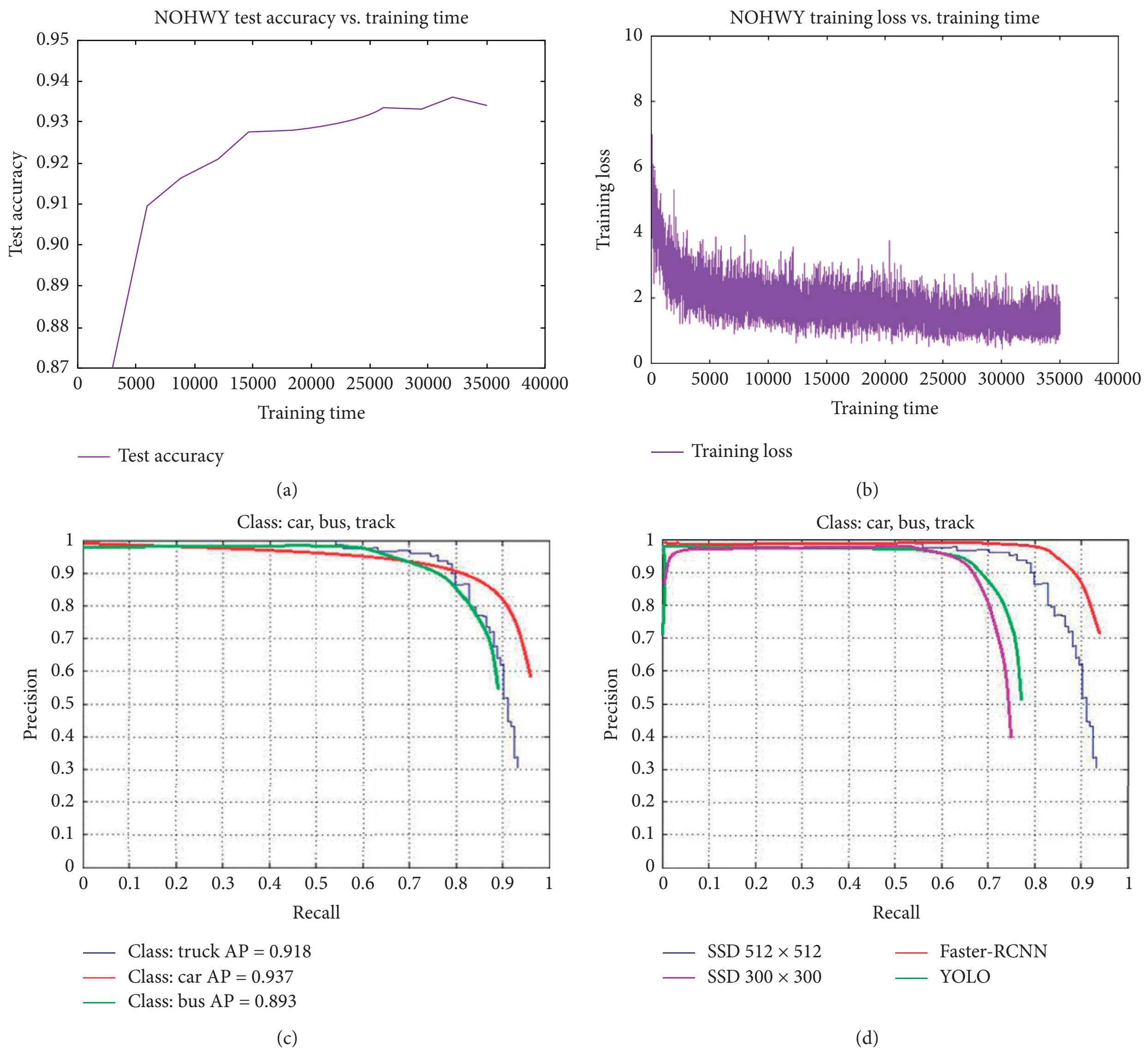


Figure 9: Curves of training results and testing results on NOHWY dataset. (a) Test accuracy-training time curves of NOHWY dataset trained by the improved SSD 512 ∗ 512 model. (b) Training loss-training time curves of NOHWY dataset trained by the improved SSD 512 ∗ 512 model. (c) Precision-Recall curves of the SSD 512 ∗ 512 model detection algorithms. AP scores were used to describe the vehicle detection models' performance. (d) Precision-Recall cures on the testing set of the NOHWY dataset.

output of car, bus, and truck in the KITTI dataset. In order to analyze the adaptability of the datasets, we draw a Precision-Recall curve for each dataset to observe the AP value of independent categories. Each type of AP can be used as an indicator to evaluate the dataset. As shown in Figure 10, the AP values performed much better in the UA-DETRAC dataset than in the in KITTI and UAV-DT dataset. However, by comparing the Precision-Recall curves of KITTI, UA-DETRAC, UAV-DT, and NOHWY dataset in Figure 9, it is found that the mAP of NOHWY dataset is the best, which shows that the improved SSD 512×512 model has a good adaptability on NOHWY dataset. The reason for this result may be that our dataset contains high-resolution samples, sample collection methods based on real-time monitoring, different target labeling methods, and larger geographical transfer problem in other datasets.

3.2. Results and Discussion of Vehicle Tracking

3.2.1. Results. To verify the effectiveness of our proposed vehicle tracking methods, we applied the methods aforementioned in Section 2.2 to the Hangzhou expressway for testing. In the testing process, six PTZ cameras installed on the Hangzhou Expressway were selected to capture 20 video sequences in different climatic conditions corresponding to more than 28,000 frames recorded at different times, and the capture rate of PTZ camera was 32 frames per second. Among them, we tested our multi-target detection and tracking algorithm with four video sequences in rainy days, four video sequences in foggy days, four video sequences in sunny days, four video sequences in cloudy days, and four video sequences at night. Previews of the five types of detection and tracking results are shown in Figure 11.

Table 2: Comparison of datasets performance under different conditions.

	KITTI	UAV-DT	UA-DETRAC	NOHWY
Acquisition equipment/year	Unmanned vehicle/2012	Unmanned aerial vehicle/2018	Cannon EOS 550D camera/2016	PTZ cameras/no
Image resolution	1242×375	1080×540	960×540	1920×1080
Images	14,999	80,000	140,000	7,849
Annotation categories	8	3	3	3
Multiple scene types	Yes	Yes	Yes	Yes
Multiple views	Yes	Yes	Yes	Yes
Multiple times of day	No	Yes	Yes	Yes

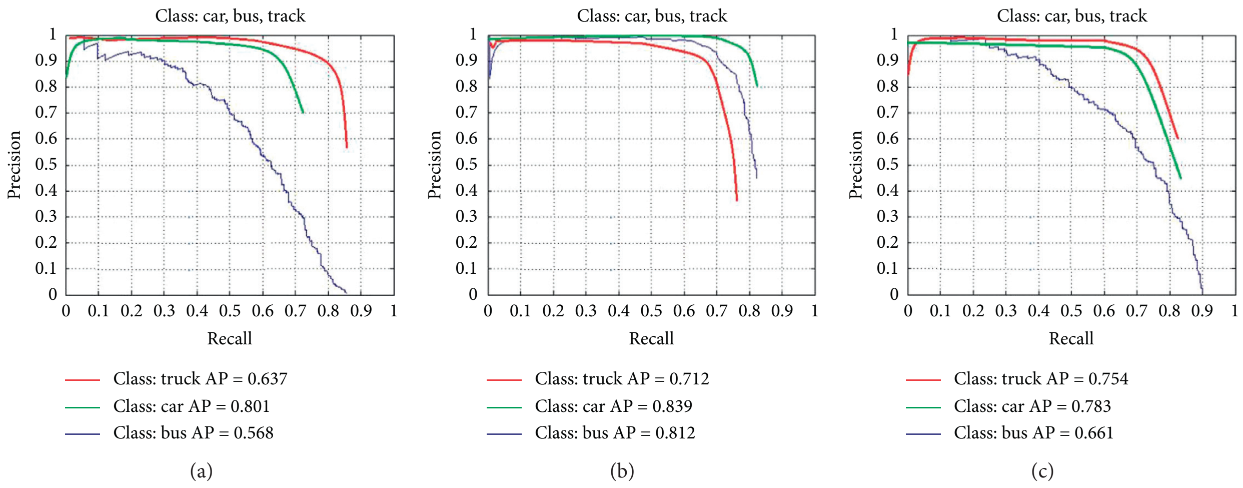


Figure 10: Precision-Recall curves of the improved SSD 512×512 method on KITTI/UA-DETRAC/UAV-DT dataset. The legend presents the AP score of each category, which can describe the performance of the detection methods. (a) KITTI. (b) UA-DETRAC. (c) UAV-DT.

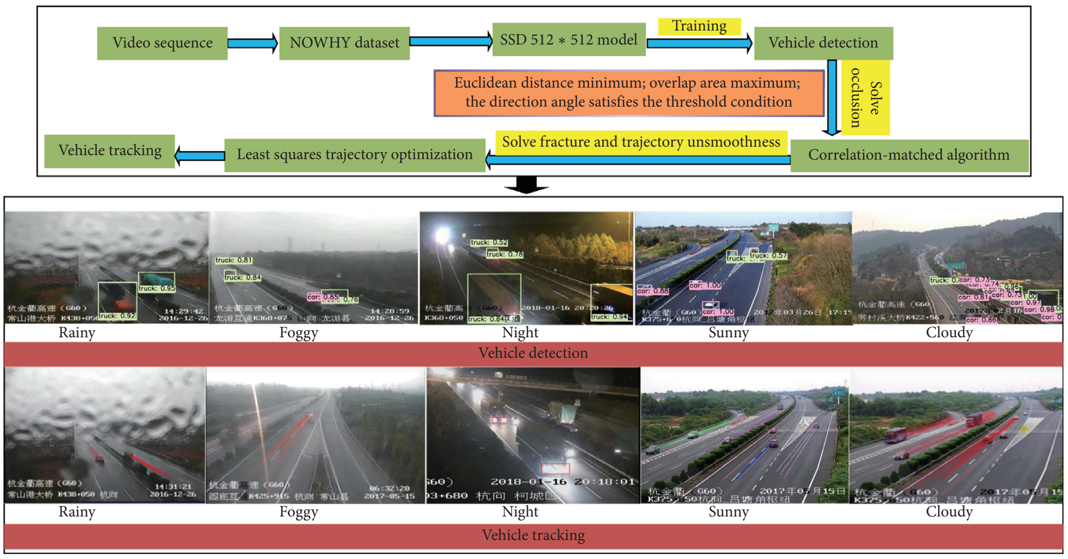

Video sequence
NOWHY dataset
SSD 512 * 512 model
Training
Vehicle detection
Euclidean distance minimum; overlap area maximum; the direction angle satisfies the threshold condition
Solve occlusion
Vehicle tracking
Least squares trajectory optimization
Solve fracture and trajectory unsmoothness
Correlation-matched algorithm
Rainy
Foggy
Night
Sunny
Cloudy
Vehicle detection
Vehicle tracking

Figure 11: Proposed methodology for vehicle tracking is used to track the video sequences under different climatic conditions.

As shown in Figure 11, firstly, we constructed a new dataset, which mainly includes three types of targets: car, bus, and truck on the expressway. Secondly, we trained our data set by an improved SSD 512×512 model. After testing the model, we obtained the result of vehicle classification and vehicle location information, which mainly includes four parameters, that is, vehicle category and the vehicle bounding boxes' height, width, and the coordinate values of the center point. Thirdly, we designed a correlation-matched algorithm for vehicle tracking, which was designed to focus on the characteristics of approximate linear motion of expressway vehicles. It is found that the trajectory of the same target is the shortest in two adjacent frames and the overlap area of the same vehicles' bounding box is the largest in the two adjacent frames, so the coordinates of the center point in the bounding box of the vehicle detected in the second step were connected. However, the connection must meet two conditions. One is to determine whether they are the same target (formula 1 and formula 2), and the other is to determine whether the trajectory connection errors caused by the same type of targets occlude each other or are running in parallel (formula 3 and formula 4). Therefore, after the correlation-matched algorithm, we obtained the trajectory points of the moving target and the trajectory connecting lines. Fourthly, we developed a trajectory optimization algorithm based on the least squares method, which smoothed the trajectory obtained in the third step and solved the problem of the trajectory unsmooth in the vehicle tracking. In the trajectory smoothing algorithm, we used the piecewise smoothing method to smooth every ten trajectory points, which can ensure the accuracy of the trajectory to the greatest extent. Finally, our tracking method in Figure 11 can achieve 26 frames per second (FPS).

3.2.2. Discussion. The correlation-matched algorithm for vehicle tracking we designed is used for the real-time monitoring system of expressway vehicles. Therefore, we compared four recent tracking algorithms including correlated kernel filters (Kalman [22], KCF [23]), online methods (MDP [17]), and batch based methods (IOUT [18]). In the testing process, four videos with a resolution of 1920×1080 in different camera view obtained by the expressway monitoring system provided by Hangzhou Huijing Technology Co., Ltd. were used in our experiment. We used multiple parameters as evaluation indicators to verify the quality of the tracking algorithm. These include FM, IDS, MOTA, and FPS (frames per second). Among them, FM indicates the total number of times the target's trajectory was fragmented during the tracking process. IDS means the total number of times the target's trajectory changed the vehicle category. MOTA represents multiple object tracking accuracy; in fact, it is a final accuracy that combines missing targets, false alarm rates, and wrong conversions of categories.

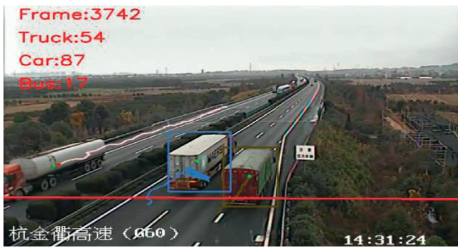
The further experiments were conducted to compare the efficiency of different tracking algorithms. As shown in Table 3, we run the tracking methods on the testing videos of the NOHWY dataset on the machine with a NVIDIA RTX 2080TI GPU. It should be emphasized that the tracking results in Table 3 are based on the accuracy of correct detection, that is, the statistical result in Table 3 do not include detection errors, but only the tracking results of the vehicles detected by the SSD 512×512 model. Due to the lack of appearance information, IOUT method only using vehicles' position information had the lower accuracy and lower IDs and FM. Moreover, MDP with SSD 512×512 had the best FPS value and the worst MOTA score among all the combinations. It is because MDP method establishes a tracking model for each vehicle, which greatly increases the FPS, but the MOTA value is lower for the reason of vehicle characteristic information absence. Meanwhile, KCF and Kalman filter method need to predict the possible candidate regions of the vehicle position, so the KCF and Kalman method perform poorly on complex background interference, similar object occlusion, which leads to unsatisfactory values of FM, IDS, and MOTA. In contrast, our tracking algorithm does not need to predict the vehicle position but directly connects the center points of the bounding box of the detection target, which greatly improves the tracking value of FPS. However, the premise of the connection is to meet the constraints we designed for occlusion, deformation, and vehicle scale change. Therefore, our tracking algorithm had the best value of FM, IDS, and MOTA, which is far superior to the other all the combinations.

3.3. Results and Discussion of Vehicle Counting. When vehicle tracking is completed, the method of detection line is usually used to achieve vehicle counting, including single detection line and double detection line. The detection line method is that when the trajectory crosses the detection line, the counter is incremented by one. However, this method can only be detected when the trajectory reaches a specific location; especially if the trajectory is discontinuous, the counting effect will be poor. In contrast, our counting algorithm is designed based on the change of the vehicles' movement trajectory point in a period of time, which greatly improves the accuracy of counting. In Section 2.3, our counting algorithm counts vehicles by establishing a coordinate system for the detection area and judging the positive and negative of the first and last trajectory points of the vehicle in the detection area. As shown in Figure 12, the height, visual angle, and climate conditions of the cameras are different. In our experiments, we implemented the correlation-matched algorithm (Figure 12(a)), optimization of the trajectory by least squares method (Figure 12(c)), detection line counting method (Figure 12(b)), and our counting method (Figure 12(d)). In the implementation of the method, Figures 12(a) and 12(c) use the traditional single line detection method to count vehicles, which has an error count problem such as omission. Compared with Figure 12(a), Figure 12(c) adds the least square method to optimize the trajectory after the correlation-matched algorithm to ensure the smoothness of the trajectory for multi-vehicle tracking. Meanwhile, both Figure 12(c) and Figure 12(d) use our tracking algorithm, which not only solves the tracking discontinuity caused by occlusion based on smooth trajectory but also has significant effect on small vehicles tracking under high-perspective scenario. Compared with the single detection line counting method of Figure 12(c) and the double detection line counting method of Figure 12(b), our counting method in Figure 12(d) is the most advanced method, and its counting time is 25 frames per second (FPS). Moreover, the error rate of our counting method is the lowest.

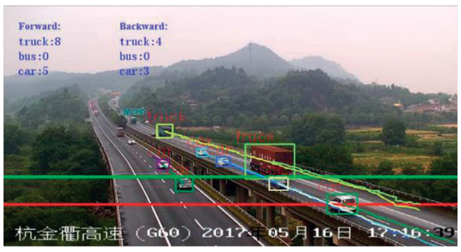
To further demonstrate the robustness of our proposed multi-vehicle counting method, we tested five expressway video sequences in different climatic conditions. Table 4 shows the results of the counting algorithm under different climatic conditions, in which the accuracy refers to the average accuracy.However, it should be noted that the

Table 3: Vehicle tracking results and quantitative comparison for different methods.

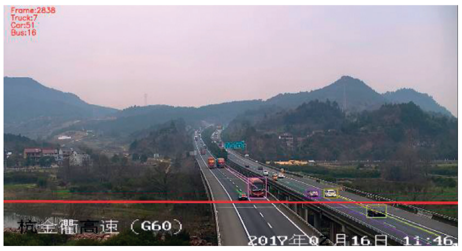
Methods		Test video							
			Camera view			FM	IDS	MOTA	FPS
Detector	Tracker	Total frames	Front view	Side view	Top view				
SSD 512 × 512	KCF [23]	4830	Yes	No	No	1084	597	87.8	17
SSD 512 × 512	Kalman [22]	3699	No	Yes	No	1756	629	83.2	14
SSD 512 × 512	MDP [17]	5124	No	No	Yes	2539	1332	74.5	52
SSD 512 × 512	IOUT [18]	3876	No	Yes	No	1875	931	76.6	7
SSD 512 × 512	Ours	5421	Yes	No	No	52	101	99.8	26



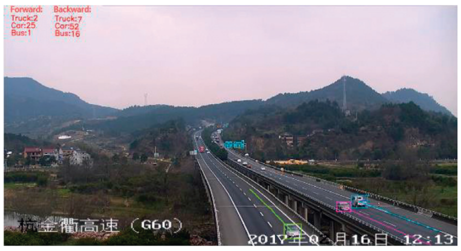
(a)



(b)



(c)



(d)

Figure 12: Illustrations of vehicle tracking and counting result on expressway video sequence. (a) Experimental results of correlation-matched algorithm and single detection line counting method under low-perspective scenario. (b) Experimental results of the correlation-matched algorithm and the double detection line (forward direction and backward direction) counting methods under low-perspective scenario. (c) Experimental results of our tracking algorithm and single detection line counting method. Among them, our tracking algorithm uses the least squares method to optimize the trajectory after the correlation-matched algorithm under challenging high-perspective scenario. (d) The experimental results of our tracking and counting methods. Among them, our counting method defines the forward and backward directions mentioned in Section 2.3 under challenging high-perspective scenario.

Table 4: Vehicle counting results and quantitative evaluation for test video.

			Counting results				
			Background				
Expressway videos	Scenarios	Camera height (meters)	Fixed	Moving	Total frames	Counted vehicles/total vehicles	Accuracy (%)
TEST-VIDEO-1	Sunny	17	Yes	No	2864	9813/9872	99.4
TEST-VIDEO-2	Rainy	15	Yes	No	2179	9024/9143	98.7
TEST-VIDEO-3	Cloudy	16	Yes	No	2430	14479/14581	99.3
TEST-VIDEO-4	Foggy	15	Yes	No	2099	6226/6340	98.2
TEST-VIDEO-5	Night	15	Yes	No	2510	8895/8976	99.1

counting results in Table 4 are based on the detection results. Therefore, the statistics do not include detection errors. It can be said that the count statistics (including tracking error and count error) in Table 4 are for the vehicle detection results of the SSD 512 × 512 model. We compared the detailed results of different scenarios and different camera heights and evaluated the quantitative performance of fixed background. In addition, total vehicles in Table 4 represents the total number of vehicles detected by the SSD 512×512 model. As reflected by the results, the accuracy of our counting method is higher than 90% under the condition of low camera height and good weather conditions. Different from the single detection line method and the double detection line method, which only record the instantaneous changes of the vehicle motion, our counting method records the changes of vehicles in a period of time (within the region), which greatly improved the accuracy of counting.

Table 5: Comparison of vehicle counting performance and evaluations between related works.

Method	Platform	Detector	Tracker	FPS	Accuracy (%)
Liu et al. [2]	Inter core i9-9900X 10 Core/3.50 GHz/19.25 MB CPU	Image processing method (updated background image and background subtraction)	No	10	91.5
Abdelwahab [31]	Inter core i9-9900X 10 Core/3.50 GHz/19.25 MB CPU	Image processing method (create background model)	No	8	90.7
Yang and Su [32]	Inter core i9-9900X 10 Core/3.50 GHz/19.25 MB CPU	Background subtraction method (low-rank + sparse)	Kalman filter algorithm	12	92.2
Abdelwahab [16]	NVIDIA RTX 2080TI GPU	DNN	KLT	15	90.4
Rosas-Arias et al. [27]	2.0 GHz Intel CPU	Incremental PCA	No	24	92.9
Ours	NVIDIA RTX 2080TI GPU	SSD 512×512	Correlation-matched algorithm	25	93.1

Accuracy, mean average accuracy; FPS, frame per second.

3.4. Comparative Studies. Vehicle counting of expressway is of great significance to traffic management. To further prove the counting method we designed in this paper, a series of experiments were conducted on our dataset to compare our counting method with the state-of-the-art vehicle counting methods. Liu et al. [2], Abdelwahab [31], and Rosas-Arias et al. [27] applied image processing method such as updated background image, background subtraction, and Incremental PCA to detect vehicles. Meanwhile, vehicle tracking was not used for vehicle counting in the three methods mentioned in [2, 27, 31]. Moreover, Yang et al. [32] adopted background subtraction method for vehicle detection and Kalman filter algorithm for vehicle tracking to achieve vehicle counting. However, vehicle counting employed Deep Neural Networks (DNN) for vehicle detection and KLT tracker for vehicle tracking were proposed by Abdelwahab [16]. Compared with the background subtraction method in image processing to detect vehicles, the accuracy of our deep learning-based detection method has been greatly improved. On the other hand, compared with KLT [16] and Kalman filter tracking algorithms [32], our proposed tracking algorithm does not need to predict the vehicle position. Instead, it considers the occlusion, deformation, and other problems caused by the vehicle movement, in which constraints have greatly improved tracking efficiency and accuracy. As reflected by Table 5, we summarized the related works in detail as much as possible by their performance in term of platform, detector, tracker, FPS, and accuracy used. In a word, our counting method is superior to the method of reference mentioned in Table 5, which is mainly reflected in the improvement of the multi-vehicle detection efficiency and the superiority of the tracking algorithm performance, especially in terms of counting speed and counting accuracy.

4. Conclusions

We propose a new vehicle counting method based on multi-vehicle detection and tracking for expressway video sequence. As a result of the particularity of monitoring visual angle and vehicle operation mode in expressway video sequence, we constructed a new expressway vehicle dataset (NOHWY) including vehicle classification and labeling rule definitions. Meanwhile, we employed the SSD 512×512 deep learning method to train and test our dataset, which greatly improves the accuracy of multi-vehicle detection. In order to further validate our detection method, we compared the performance of different deep learning methods on different datasets. The experimental results show that our vehicle detection method had higher detection efficiency and accuracy. Moreover, unlike the traditional single-target tracking method, we proposed a correlation-matched algorithm for multi-vehicle tracking based on the change of time and space position when the vehicle is driving on the expressway. Meanwhile, a trajectory optimization method based on least squares method is proposed to ensure the smoothness of the tracking trajectory. In both study algorithms, the proposed multivehicle tracking method performed better than the state-of-the-art approaches. Finally, we designed a new vehicle counting method, which can accurately count the vehicles on the expressway according to the driving direction. The comparative analysis demonstrated that the proposed vehicle counting method for expressway can obtain more than 93% accuracy and 25 FPS speed on vehicle counting based on vehicle detection and vehicle tracking.

Future research work will design new algorithms for vehicle behavior analysis and traffic incident detection based on the approach and experimental results in this paper.

Data Availability

The original video and images of the expressway in this study were provided by Hangzhou Jinghui Technology Co., Ltd. In order to ensure the confidentiality of the subsequent research results, the dataset constructed by us in this paper is not publicly available.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors are very grateful to Hangzhou Huijing Technology Co., Ltd. for providing expressway videos for this research. This research was funded by the Science and Technology Project in Qinghai Province (no. 2017-ZJ-717), the National Natural Science Foundation of China (Grant no. 61572083), and the Youth Fund Project of Qinghai University (no. 2019-QGY-15).

References

[1] W. JiankaiT and S. Agachai, "Automatic freeway incident detection for free flow conditions: a vehicle reidentification based approach using image data from sparsely distributed video cameras," *Mathematical Problems in Engineering*, vol. 2015, Article ID 102380, 13 pages, 2015.

[2] F. Liu, Z. Zeng, and R. Jiang, "A video-based real-time adaptive vehicle-counting system for urban roads," *PLoS One*, vol. 12, Article ID e0186098, 2017.

[3] Z. Chengming and L. Wei, "Background updating algorithm based on classification in complex scene," *Journal of Computer Applications*, vol. 28, pp. 2274–2277, 2008.

[4] G. Deng and K. Guo, "Self-adaptive background modeling research based on change detection and area training," in *Proceedings of the IEEE Workshop on Electronics*, Ottawa, Canada, May 2014.

[5] M. Wang, G. Huang, and X. Da, "A new interframe difference algorithm for moving target detection," in *Proceedings of the 2010 3rd International Congress on Image and Signal Processing*, Yantai, China, October 2010.

[6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," *Lecture Notes in Computer Science*, vol. 10, pp. 25–36, 2004.

[7] J. Ankang, G. Yuanbo, Y. Ziwei, L. Tao, and M. Jing, "HeteMSD: A big data analytics framework for targeted cyber-attacks detection using heterogeneous multisource data," *Security and Communication Networks*, vol. 2019, Article ID 5483918, 9 pages, 2019.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate target detection and semantic segmentation," in *Proceedings od the IEEE Conference Computer Vision Pattern Recognition (CVPR)*, pp. 580–587, Columbus, OH, USA, June 2014.

[9] H. Kai, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[10] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Araucano Park, Chile, December 2015.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[12] K. M. He, G. Gkioxari, D. Piotr, and G. Ross, "Mask R-CNN computer vision and pattern recognition," 2018, https://arxiv.org/abs/1703.06870.

[13] L. Wei, A. Dragomir, E. Dumitru, and S. Christian, "S. S. D. Single shot multibox detector," in *Proceedings of the 14th European Conference Computer Vision—ECCV 2016*, pp. 21–27, Amsterdam, Netherlands, October 2016.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time target detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1804–1812, Las Vegas, NV, USA, June 2016.

[15] Z. Chen, T. Zhang, and C. Ouyang, "End-to-end airplane detection using transfer learning in remote sensing images," *Remote Sensing*, vol. 10, no. 1, pp. 139–152, 2018.

[16] M. A. Abdelwahab, "Accurate vehicle counting approach based on deep neural networks," in *Proceedings of the IEEE International Conference on Innovative Trends in Computer Engineering (ITCE)*, Aswan, Egypt, February 2019.

[17] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: online multi-object tracking by decision making," in *Proceedings of the Computer Vision (ICCV)*, pp. 4705–4713, Santiago, Chile, December, 2015.

[18] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image imformation," in *Proceedings of the Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy, September 2017.

[19] M. Danelljan, G. Hager, and F. S. Khan, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4310–4318, Santiago, Chile, December 2015.

[20] R. Ding, M. Yu, H. Oh, and W.-H. Chen, "New multiple-target tracking strategy using domain knowledge and optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 4, pp. 605–616, 2017.

[21] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 4, pp. 271–288, 1998.

[22] Z. Zhang and J. Zhang, "A new real-time eye tracking based on nonlinear unscented Kalman filter for monitoring driver fatigue," *Journal of Control Theory and Applications*, vol. 8, no. 2, pp. 181–188, 2010.

[23] S. Battiato, G. M. Farinella, A. Furnari, G. Puglisi, A. Snijders, and J. Spiekstra, "An integrated system for vehicle tracking and classification," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7263–7275, 2015.

[24] Tzutalin, "LabelImg git code(2015), http://github.com/tzutalin/labelImg.

[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu, HI, USA, June 2017.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional network for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[27] L. Rosas-Arias, J. Portillo-Portillo, A. Hernandez-Suarez et al., "Vehicle counting in video sequences: an incremental subspace learning approach," *Sensors*, vol. 19, no. 13, p. 2848, 2019.

[28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 16–21, Providence, RI, USA, June 2012.

[29] D. Du, Y. Qi, H. Yu et al., "The unmanned aerial vehicle benchmark: object detection and tracking," 2018, https://arxiv.org/abs/1804.00518.

[30] L. Wen, D. Du, Z. Cai et al., "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," 2016, https://arxiv.org/abs/1511.04136.

[31] M. A. Abdelwahab, "Fast Approach for efficient vehicle counting," *Electronics Letters*, vol. 55, no. 1, pp. 20–22, 2019.

[32] H. Yang and S. Qu, "Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition," *IET Intelligent Transport Systems*, vol. 12, no. 1, pp. 75–85, 2017.