Traffic incident detection based on a global trajectory spatiotemporal map

Haoxiang Liang¹ · Huansheng Song¹ · Xu Yun¹ · Shijie Sun¹ · Yingxuan Wang¹ · Zhaoyang Zhang¹

Received: 15 July 2021 / Accepted: 26 November 2021 / Published online: 19 December 2021
© The Author(s) 2021

Abstract

Traffic incidents endanger the smooth running of vehicles. Congestion caused by traffic incidents has caused a waste of time and fuel and seriously affected transportation efficiency. At present, most methods use manual judgment or image features to detect traffic incidents, but these methods lack timeliness, leading to secondary incidents. For dangerous road sections such as ramp-free and long downhills, this paper proposes an algorithm to quickly detect traffic incidents based on a spatiotemporal map of vehicle trajectories. First, a vehicle dataset from the monitoring perspective is constructed, and an improved YOLOv4 detection algorithm is used to detect images organized as batches. Based on the detection result, the multi-object tracking method of vehicle speed prediction in key frames is used to obtain the vehicle trajectory. Then according to the vehicle trajectory obtained in a single scene, the vehicle trajectory is reidentified and associated in the continuous monitoring scene to construct a long-distance vehicle trajectory spatiotemporal map. Finally, according to the distribution and generation status of the trajectory in the spatiotemporal map, traffic incidents such as vehicle parking, vehicle speeding, and vehicle congestion are analyzed. Experimental results show that the proposed method greatly increases the speed of vehicle detection and tracking and obtains high mAP, MOTA, and MOTP indicators. The global spatiotemporal map constructed by trajectory reidentification can achieve high detection rates for traffic incidents, reduce the average elapsed time, and avoid the problems of the inaccuracy of analyzing image features.

Keywords Automatic incident detection · Spatiotemporal map · Vehicle detection and tracking · Intelligent transportation system · Surveillance video

Introduction

Real-time detection of traffic incidents can reduce congestion caused by traffic incidents. Traffic incident detection includes abnormal parking, traffic congestion, and retrograde vehicles. Every 7 minutes of delay in a traffic incident increases the traffic jam by 1 mile [1], which increases the possibility of secondary incidents. Therefore, managers must detect and deal with the incident in time. Pattern recognition and genetic algorithms have been used to quickly detect traffic incidents. However, a national survey [2] found that because Automatic Incident Detection (AID) has the characteristics of complexity, a high false alarm rate, and delay, 90% of the survey respondents believe that such methods are not suitable

for use at present or in the future. Therefore, it is necessary to redesign the AID algorithms to overcome the above problems. Traditional AID algorithms mostly use hardware devices on highways to detect traffic incidents. Data utilized includes radar, loop detector, floating car, and cameras. The existing video-based incident detection methods mainly rely on semantic information in images. In the 2018 and 2019 Ai CITY challenges, Naphade et al. [3,4] increased the detection speed of traffic incidents to 3–10 seconds, and the use of video images on highways can significantly improve the detection speed and accuracy of traffic incidents [5].

Using video surveillance data to detect traffic incidents in real time [6], compared with traditional pressure sensors [7] and lidar [8], greatly improves the detection accuracy and response time. The use of images to obtain real-time traffic status can be divided into three methods: (a) detection, (b) motion, and (c) holistic. Detection-based methods use video frames to detect vehicles. In [9], Kalman filtering was used to estimate the background of the traffic

✉ Huansheng Song
hshsongtrb@163.com; hshsong@chd.edu.cn

¹ School of Information Engineering, Chang'an University,
Middle Section of Nan Erhuan Road, Xi'an, China

scene and calculate the vehicle density. Reference [10] proposed Region-Convolutional Neural Networks (R-CNNs) to improve the accuracy of traffic density calculations. Reference [11] used two CNN variants (counting CNN and hydra CNN) to count vehicles and predict traffic density. However, the above methods have difficulty in accurately counting the density of vehicles with low resolution and severe occlusion of traffic videos. The motion method was used to estimate the traffic status, that is, track the vehicle and calculate the traffic flow. Reference [12] used the motion of the vehicle to extract motion parameters for vehicle counting, but if the monitoring angle is low and the vehicle flashes past in the scene, it is difficult to obtain the trajectory of the vehicle, and thus vehicle counting cannot be achieved. Using a holistic method to estimate the traffic status, [13] performed linear transformation on the pixels in the frame to obtain different densities of different objects to judge congestion. Reference [14] used a spatiotemporal Gabor filter to extract the range of vehicles in the scene and separated congestion into different types. However, this method may fail when monitoring a wide field of view.

Using deep learning for traffic incident detection, some work results have been achieved. Reference [15] used convolutional neural networks, [16] used Deep Belief Networks (DBNs), and [17] used Long Short-Term Memory networks (LSTMs) for traffic incident detection. These methods all require a large number of data samples and manual annotations, which results in poor portability and results.

Based on object detection and tracking in surveillance scenes, this paper designed a method for traffic incident detection using a global trajectory spatiotemporal map, as shown in Fig. 1. First, the method reads the surveillance video stream to detect traffic objects and add batch image processing to You Only Look Once vision 4 (YOLOv4) to improve detection efficiency. Second, we use a key frame multi-object association method guided by the vehicle detection results of the convolutional network to complete Multi-Object Tracking (MOT) and obtain accurate vehicle trajectories. According to the vehicle trajectory and vehicle position in each scene, the vehicle is reidentified across scenes to associate the same vehicle between different scenes. We use the results of vehicle reidentification to generate a global spatiotemporal map of the trajectory under the current road segment, thereby analyzing traffic incidents such as traffic congestion, vehicle speeding, and illegal parking using the trajectory state. In this way, effective incident detection can be performed on traffic scenes captured by continuous cameras without ramps.

The main contributions of this research affect theories and applications. For theoretical contributions, first, since the establishment of the global spatiotemporal map in this paper involves multiple cameras, this research designs a batch detection method for the YOLOv4 network, adding

an encode layer and decode layer to achieve accurate vehicle detection results for multiple videos, and improves the processing efficiency of the video stream. Second, this research proposes a multi-object tracking algorithm using vehicle information from key video frames. The tracking algorithm uses the vehicle speed to predict the position of the vehicle and realizes the matching and association of the vehicle trajectories with high-level tracking accuracy and precision. The method greatly reduced the number of video frames required for vehicle tracking, improved the tracking speed, and ensured the continuous and stable generation of the vehicle trajectory. Finally, a global spatiotemporal map generation method combining vehicle similarity and spatial position is proposed. The loss function associated with the same vehicle under multiple cameras is designed to ensure that the same vehicle is correctly identified under different cameras.

For application contributions, first, to meet the needs of traffic incident analysis in different traffic environments, this research proposes a vehicle object dataset construction idea for a problem in which the vehicle captured by the surveillance camera is severely deformed due to vehicle motion. The dataset fully considers the surveillance video scenes of day, tunnel, and night under different perspectives and has 59,672 instances of labeling. Among them, the labeling of larger and smaller objects is especially considered. This can provide a reference for the organization and construction of vehicle datasets. Second, in the vehicle tracking experiment, a multi-object tracking dataset was published. The tracking dataset has 238 manually labeled vehicle trajectories and a total of 27,544 bounding boxes, which can be used for the evaluation of various MOT algorithms. Finally, this research proposes a traffic incident detection method using global spatiotemporal maps. This method can give a stable and correct judgment of a variety of traffic incidents. Global spatiotemporal maps can be widely used in transportation applications such as the study of car-following models and the prediction of road flow and frequent incidents, to provide assistance for traffic management and control.

This article will be described in the following sections. In second section delineates the related works of vehicle detection and tracking. In third section introduces the technology in detail, fourth section describes the establishment of the global spatiotemporal map and the method of traffic incident judgment. In fifth section presents the experimental results and compares them with other algorithms, and describes the self-made tracking dataset. In sixth section discusses the qualitative analysis of some results, and explains the limitations of the algorithms, then gives directions for future work. Finally, seventh section summarizes the entire article.

Fig. 1 Using trajectory spatiotemporal map for traffic incident detection

Related work

Vehicle detection from monitoring perspective

In recent years, the development of deep convolutional networks has greatly improved the performance of object detection and classification. The number of network layers and functional units has greatly increased. From image classification tasks to further object detection tasks, two-step detection algorithms represented by R-CNN [18] and Fast R-CNN [19] have emerged. However, the two-step algorithm is more sensitive to changes in object scale. In traffic monitoring scenarios, it is obvious that when the same vehicle moves, it will have different object sizes according to different positions in the image, and the deep feature representations of large and small vehicles in the convolutional network are very different [20]. Since a Region of Interest (RoI) pooling layer is used in the two-step detection network, each region proposal is represented by a fixed-size feature vector, and the proposed region is divided into $H \times W$ subwindows. Max pooling is used to extract a value in each window. If the proposed region is smaller than the set $H \times W$ size, then elements in some positions of the proposed region will be copied to fill the proposed area to the $H \times W$ size. In this way, the image of the small target corresponding to the proposed region is destroyed. In the network training, the structure filled in this way is used for forward propagation. Errors are accumulated, resulting in incorrect training, and the object may not be detected correctly.

To solve the problem of scale sensitivity of convolutional neural networks, most of the existing solutions use an image pyramid structure to make the input image suitable for each scale of the network. This increases the amount of calcula-

tion and is unacceptable for real-time object detection tasks. SI Net [20] designed an RoI pooling layer that uses context awareness to merge each proposal region on different layers into a fixed-size feature vector so that the small proposal area is deconvolutionally amplified using a bilinear kernel to achieve a good representation. In the single-step detection algorithm, the Single Shot multibox Detector (SSD) [21] network uses image pyramids to extract features of objects from different levels, uses high-resolution images to detect small objects, and uses low-resolution images to detect large objects. However, the high-resolution images obtained using image pyramid magnification lose a significant amount of semantic information, and small targets cannot be detected accurately. The YOLOv4 [22] network joined the Spatial Pyramid Pooling (SPP) [23] module, using multiple maximum pooling layers, effectively increasing the acceptance range of backbone features, separating the most important context features, and then fusing feature maps of different scales. This enriches the expression ability of feature maps and improves the accuracy of detecting objects of different sizes. In addition, YOLOv4 draws on the ideas of the Path Aggregation Network (PANet) [24] using a Feature Pyramid Network (FPN) layer to convey strong semantic features from top to bottom, while the feature pyramid conveys strong positioning features from bottom to top, which enhances the ability to extract features. To enrich the instance pose, YOLOv4 uses Cutmix [25] for data enhancement, cutting out some areas in the image and randomly filling the pixel values of same-size areas in other images. This enhances the robustness and generalization ability of the detection model. As shown in Fig. 2, the truck is driving toward the camera. Using YOLOv4 to perform object detection, it can be seen that the response of the truck object in the feature map moves

from small to large. This shows that the above improvements in YOLOv4 play a key role in detecting objects of different scales and provide a solution to the problem of scale changes caused by vehicle movement in traffic scenes.

In addition, Chen et al. [26] recently converted large- and medium-sized instances into medium- and small-sized instances by scaling and splicing images. This increased the number of small instances and improved the quality of instances. Scaled-YOLOv4 [27] is designed on the basis of the YOLO series network; uses the network scaling method to adjust the depth, width, resolution, and results of the network; and proposes the YOLOv4-Cross Stage Partial (CSP), YOLOv4-tiny, and YOLOv4-large models. These models improved detection performance, balanced computational complexity, and memory usage, and can efficiently process data on hardware platforms with different performances. You Only Look One-level Feature (YOLOF) [28] canceled the FPN structure, adopted a dilated encoder and uniform matching modules, designed a deep network that only uses single-level features for detection, and added random movement to the image. This alleviates the problem of insufficient matching between the bounding boxes and anchors. Although YOLOF is weaker than YOLOv4 [22] in small-object detection, it improves the inference speed and achieves a balance between speed and accuracy. Supervised comparative learning has made progress in image processing. DetCo [29] established the contrast loss between the partial and the entire image. The partial image patch removes the contextual information of the image, which enables better comparative learning and obtains competitive object detection results.

Multi-object tracking

MOT estimates the motion of all objects in a scene. Based on the detection results, MOT needs to continuously associate the same object in different frames to form a continuous trajectory. In the real-time online MOT algorithm, because the global optimal solution cannot be obtained, when the object is blocked or the detection result is incorrect, the associated trajectory may be interrupted, resulting in multiple trajectory fragments. To solve the above problems, the Multiple Hypothesis Tracking (MHT) [30] algorithm is used to retain all hypotheses of the tracked object and eliminate the current uncertainty from the subsequent observation data. Using the Joint Probabilistic Data Association Filter (JPDAF) [31] method, the associated probability between the observation data and each object is calculated, and a single-state hypothesis is generated by weighting and the associated probability. Under ideal conditions, MHT can obtain the optimal solution for data association and can handle the addition of new objects and the deletion of old objects. However, when the number of objects increases, both MHT and JPDA require

complex calculations, which makes it difficult to ensure real-time performance.

To ensure industrial practicability, [32] proposed the Simple Online and Real-time Tracking (SORT) tracking algorithm. This algorithm uses a Kalman filter to predict the object position and a Hungarian algorithm to match the trajectory with the object one by one. The SORT algorithm can simply achieve the effect of real-time tracking, but when the movement of the object has a strong uncertainty, the Kalman filter cannot give an accurate prediction position, which leads to poor performance. To solve this problem, [33] proposed the Deep-SORT algorithm. Based on the SORT algorithm, Deep-SORT added a matching cascade mechanism to focus on the movement and appearance information of the object to ensure that the trajectory matched correctly in consecutive frames. Based on Deep-SORT, MOTDT [34] uses neural networks to solve unreliable detection and occlusion problems and designs a trajectory scoring function to make accurate choices from the candidates. Based on MOTDT, the Jointly learns the Detector and Embedding model (JDE) [35] extracts the embedding vector in the feature map and uses triplet loss to achieve a higher tracking speed than MOTDT. However, the JDE still needs to perform object detection first and then match objects. This requires reliable object detection results; otherwise, the tracking will fail.

CenterTrack [36] unified object detection and data association, input the RGB image of the current and previous frames and the predicted heatmap image of the previous frame to the same network and then merged the image information with a bitwise adding method. CenterTrack only correlated the detection bounding boxes between two consecutive frames, which balanced the tracking speed and detection accuracy. However, it was difficult to form a long-term object association relationship, which caused a problem in tracking when the target ID switched frequently. Track to Detect and Segment (TraDeS) [37] proposed an online joint detection and tracking model that put adjacent video frames into a deep network to determine the changes in the bounding boxes. Tracking auxiliary detection was used to correct the network results of the detection. This method obtained stable and effective tracking results. Siamese MOT (SiamMOT) [38] uses an area-based multi-object tracking framework that can simultaneously complete the detection and correlation tasks in multi-object tracking. SiamMOT uses a motion model to estimate the movement of objects between two frames, thereby correlating the detected objects in the two frames, and verifies the importance of motion modeling for multi-object tracking on multiple datasets. However, the multi-object tracking algorithm using deep learning requires more computing time [39]. Future research will try to integrate new strategies with classic feature processing methods to develop real-time speed and high-precision tracking.

Fig. 2 Vehicle scale changes drastically from monitoring perspective

Therefore, on the basis of vehicle detection and tracking, algorithms are proposed in the following sections to take both speed and accuracy into consideration to ensure timely and accurate responses to traffic incident detection.

Proposed method

Vehicle dataset

Since there are few open-source and high-resolution large datasets from the monitoring perspective, we constructed a vehicle dataset, as shown in Table 1. The dataset covers 31,809 images from various perspectives and various scenes. These images meet the needs of detection and testing in different traffic scenarios and are all of 1920*1080 resolution. Three types of vehicles (Car, Truck, and Bus) are labeled in the surveillance image, and each image has 1.8759 labeled instances on average.

As shown in Fig. 3, in the dataset, three different scenes (day, tunnel, and night) are considered. According to the height of the camera, each scene is subdivided into a high viewing angle with a height of approximately 15–20 m and a low viewing angle with a height of approximately 4.5 m. A wide range of roads are photographed at a high viewing angle, and small targets within approximately 300 m from the

surveillance camera can be seen. At low viewing angles, the object deforms drastically, and the object occupies a larger area in the image. In particular, in daytime scenes, the labels are marked from dawn to darkening. In the tunnel scene, due to the limited space, when the vehicle passes through the tunnel, the vehicle undergoes drastic deformation from the monitoring perspective. Therefore, we only label vehicles that completely enter the camera's field of view to ensure the integrity of the object. In a dark night scene, because a grayscale image is collected, we consider the situation of insufficient night light and significant dizziness from headlights and mark the visible range of the vehicle target as much as possible to ensure the accuracy of the ground truth of the marked object.

Improved network structure

We use YOLOv4 to detect vehicles. To meet the needs of real-time processing and improve detection efficiency, we designed an improved YOLOv4 network structure, as shown in Fig. 4. We added a batch encode layer before the input structure of YOLOv4. This layer reads 4 frames of video stream images, stitches 2 images by row and 2 images by column to organize into a batch image we need, and resizes the batch image to 416*416 as the input to the original YOLOv4 network. After the output branch of the original YOLOv4

Table 1 Vehicle dataset information from monitoring perspective

Scenes	Number of images	Image size	Number of instances
Day	18,314	1920*1080	29,274
Tunnel	8173	1920*1080	19,802
Night	5322	1920*1080	10,596

Fig. 3 Scenes and instances in vehicle dataset

network, we added a batch decode layer that can return the detection results in the batch image to four single images for output.

In particular, in the batch encode layer, we performed data conversion to meet the input format of image annotation during training. As shown in Fig. 4, first, the four images are resized to $w * h$, and the batch image is $2w * 2h$. Therefore, the labeled position of the objects in the four images needs to be converted to the label position corresponding to a $2w * 2h$ image (YOLO-mark format). This uses Eqs. (1)–(4) to perform the conversion and satisfy the input of the batch encode layer. In Eqs. (1)–(4), x_{obj} , y_{obj} are the upper left corner of the labeled box, and w_{obj} , h_{obj} are the width and height of the labeled box. $x_{labeled}$, $y_{labeled}$, $w_{labeled}$, and $h_{labeled}$ are the label positions required by the batch encode layer.

$$x_{labeled} = \begin{cases} x_{obj}, & \text{Object belongs to photo 1 or 3} \\ x_{obj} + w, & \text{Object belongs to photo 2 or 4} \end{cases} \quad (1)$$

$$y_{labeled} = \begin{cases} y_{obj}, & \text{Object belongs to photo 1 or 2} \\ y_{obj} + h, & \text{Object belongs to photo 3 or 4} \end{cases} \quad (2)$$

$$w_{labeled} = \frac{w_{obj}}{2w} \quad (3)$$

$$h_{labeled} = \frac{h_{obj}}{2h} \quad (4)$$

In the batch decoding layer, we return the output results to four single images. We assign the detection result to its own image according to the position of the upper left corner of the bounding box in the original output of YOLOv4. When the output bounding box position exceeds the width and height of the original image, we correct the coordinate position so that the bounding box size does not exceed the range of the image. In Eqs. (5)–(8), x_{output} , y_{output} , w_{output} , and h_{output} in the original result of YOLOv4 are the upper-left corner point and the width and height coordinates of the bounding box, respectively. x_{decode} , y_{decode} , w_{decode} , and h_{decode} are the upper-left corner point and the width and height coordinates of the bounding box output from the batch decode layer.

$$x_{decode} = \begin{cases} x_{output}, & x_{output} < w \\ x_{output} - w, & x_{output} \geq w \end{cases} \quad (5)$$

$$y_{decode} = \begin{cases} y_{output}, & y_{output} < h \\ y_{output} - h, & y_{output} \geq h \end{cases} \quad (6)$$

$$w_{decode} = \begin{cases} w - x_{output}, & w_{output} + x_{output} \geq w \\ w_{output}, & \text{other} \end{cases} \quad (7)$$

$$h_{decode} = \begin{cases} h - y_{output}, & h_{output} + y_{output} \geq h \\ h_{output}, & \text{other} \end{cases} \quad (8)$$

Fig. 4 Improved YOLOv4 network structure

Key frame multi-object tracking

In the surveillance video scene, according to the vehicle detection results, the vehicles are tracked. To ensure the efficiency of tracking, we adopt a key frame tracking method using speed prediction to obtain the complete vehicle trajectory. The overall tracking process is shown in Fig. 5.

As shown in Fig. 5, the vehicle detection results of two adjacent frames are read, the Intersection over Union (IoU) overlap of the vehicle detection box in the two frames is calculated, and the association between the same vehicle in the two frames is determined. As shown in Eq. (9), suppose the range of the p th detection box in the N th frame is A_N^p , and the range of the q th detection box in the $N + 1$ th frame is A_{N+1}^q . Then, IoU is calculated as:

$$IoU(A_N^p, A_{N+1}^q) = \frac{\text{area}(A_N^p \cap A_{N+1}^q)}{\text{area}(A_N^p \cup A_{N+1}^q)} \quad (9)$$

Taking the largest value of the IoU result, it is considered that the p th object in the N th frame is related to the q th object in the $N+1$ th frame. Then, we calculate the initial speed of vehicle $\text{Speed}_{\text{init}}$ according to the above two associated detection boxes. When calculating the speed, it is necessary to convert the image coordinates of the vehicle bounding box to the world coordinate system, which uses camera calibration. This will not be discussed in detail in this article. Suppose the coordinates of the lower bottom midpoint of the p th vehicle bounding box in the N th frame and the q th vehicle bounding box in the $N + 1$ th frame converted to the world coordinate system are (U_N^p, V_N^p) , (U_{N+1}^q, V_{N+1}^q) , respectively. The video frame rate is FPS, and the calculation method of the vehicle's initial speed is shown in Eq. (10).

$$\text{Speed}_{\text{init}} = \frac{\sqrt{(U_{N+1}^q - U_N^p)^2 + (V_{N+1}^q - V_N^p)^2}}{1/\text{FPS}} \quad (10)$$

Next, we use the Lucas–Kanade optical flow method [40] to track the vehicle and use the IoU overlap to associate the same vehicle between frames to ensure the generation of vehicle trajectories. The trajectory association results can be divided into three types.

1. Trajectory matching failed. This means that there are redundant trajectories but no remaining detection boxes can be matched. At this time, we use the speed to give the predicted position of the vehicle in the frame. The method uses the vehicle speed and the time across the frame to calculate the distance traveled by the vehicle, give the predicted vehicle position, and use the IoU correlation to match the redundant trajectory. When the redundant trajectory does not match the bounding box three consecutive times, it is considered that the vehicle has driven out of the scene, and the bounding box is no longer matched to the trajectory.
2. Detection matching failed. This means that there are redundant bounding boxes but no remaining tracks to match. At this time, false detection is performed. The redundant bounding box is tracked for 3 frames. If the bounding box still does not match the existing track but can be continuously associated, then it is considered a new object, and a new track is created. Otherwise, the object is considered to be erroneously detected, and the track is deleted.
3. Object and trajectory successfully matched. At this time, the trajectory and the bounding box are matched one by one, and the detection results of the interval 5 frames can be read in. According to the principle of Eq. (10), the vehicle detection results of the $N+5$ th frame and the N th frame are calculated to update the speed of the vehicle $\text{Speed}_{\text{update}}$. In particular, if the number of nodes in the trajectory exceeds 5, then the trajectory is considered to be a valid trajectory, and the trajectory is saved. For trajectories with fewer than two nodes, the trajectory

Fig. 5 Flowchart of key frame vehicle tracking

is considered to be interference noise and is deleted. In this way, the complete trajectory of the vehicle in each independent scene is obtained.

Generation and analysis of trajectory spatiotemporal map

Global spatiotemporal map generated by trajectory reidentification

Using the global trajectory spatiotemporal map can help analyze the vehicle status and traffic incidents in continuous

scenes, as shown in Fig. 6. First, we draw the vehicle trajectory generated by a single scene onto the spatiotemporal map. The time (frame number) is used as the abscissa in the spatiotemporal map, and the distance is the ordinate. This distance refers to the distance between the vehicle position and the camera. We reidentify the trajectory generated under the independent camera to obtain a spatiotemporal map of the trajectory in the multiple continuous surveillance scenes. The trajectory reidentification of the same vehicle in different scenarios is divided into two parts: the matching of the trajectory fitting line parameters and the matching of the vehicle ROI image in the trajectory node. We connect the matched trajectories in different scenarios, as shown by the

dotted line in Fig. 6. To easily distinguish the driving status of the vehicle, we only consider one-way road surveillance video. Since the trajectories generated by the vehicle almost all have the same instantaneous speed, we fit the trajectory into a linear regression equation, expressed as the relationship of distance (y_{dis}) and time (x_{time}). Then, we put the slope a and intercept b parameters in the linear regression equation into the Hough space. At this point, trajectories generated by different vehicles correspond to a set of points in Hough spaces. Assuming that a vehicle generates n sets of trajectory points $(x_1, y_1), \dots, (x_i, y_i)$, using the least square method, the solutions of a and b can be obtained as shown in Eq. (11).

$$\begin{cases} a = \frac{\sum_{i=0}^{n-1} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=0}^{n-1} (x_i - \bar{x})^2} \\ b = \bar{y} - a\bar{x} \\ \bar{x} = \sum_{i=0}^{n-1} \frac{x_i}{n} \\ \bar{y} = \sum_{i=0}^{n-1} \frac{y_i}{n} \end{cases} \quad (11)$$

We use the above algorithm to fit the vehicle trajectory under multiple cameras involving continuous scenes. To obtain the best re-ID result, we designed $loss_{(k,j)}$, as shown in Eq. (12). Suppose the linear parameter fitted to the k th trajectory in scene S is a_S^k, b_S^k , and the linear parameter fitted to the j th trajectory in scene $S+1$ is a_{S+1}^j, b_{S+1}^j . We calculate the distance between two points according to Eq. (12) and multiply it by the weight w_1 . Meanwhile, we take the ROI area of the vehicle corresponding to the middle node of the k th trajectory in scene S and the j th trajectory in scene $S+1$: ROI_{S+1}^j, ROI_S^k , calculate the color histogram similarity $\cos(ROI_{S+1}^j, ROI_S^k)$ of the two ROI images, and then multiply it by the weight w_2 . We take w_1, w_2 as 0.75, 0.25. When $loss_{(k,j)}$ reaches the minimum value, k, j is the matching trajectory number, which is considered to be the trajectory of the same car in different scenes. At this time, we connect the j th trajectory in scene $S+1$ with the k th trajectory in the scene S , draw a continuous vehicle trajectory, complete trajectory reidentification, and construct a global trajectory spatiotemporal map for detecting traffic incidents.

$$loss_{(k,j)} = w_1 * \left(\sqrt{(a_{S+1}^j - a_S^k)^2 + (b_{S+1}^j - b_S^k)^2} \right) + w_2 * (\cos(ROI_{S+1}^j, ROI_S^k)). \quad (12)$$

This paper uses the results of camera calibration to determine the horizontal and vertical positions of the vehicle trajectory. When two cars are the same or similar in appearance, their driving trajectories may not be exactly the same. As shown in Fig. 6, in the spatiotemporal map, if two similar cars drive side by side, the trajectory will be different in the distance axis (vertical direction). If two similar cars drive in

the same lane, then the trajectory will be different in the time axis (horizontal direction). Therefore, the global spatiotemporal map matching generation algorithm takes into account the position of the vehicle trajectory and can distinguish two vehicles with the same or similar appearance.

According to the above description, we selected multiple consecutive video sequences of traffic surveillance cameras to reidentify the trajectory. As shown in Fig. 7, multiple single traffic scenes and their trajectory spatiotemporal maps are displayed, and the y -axis of the spatiotemporal map is the actual distance between the vehicle and the camera. At the bottom of Fig. 7, the global traffic spatiotemporal map created by reidentification of the trajectories of different scenes is shown. In the remote area from the surveillance camera, where the vehicles are not being tracked, we use red lines to connect the same vehicle trajectory between different scenes. The camera spacing of multiple scenes can be known from the text in the monitoring video. Using the “K924+448, K924+573, K924+698” text in the video frame, it can be calculated that the distance between the second and third surveillance cameras is 125 m and 250 m from the first camera. It can be seen that using the algorithm in this section, the trajectory of the same vehicle under different cameras can be effectively identified and connected so that the traffic state can be identified based on the complete vehicle trajectory.

Traffic incident detection

Based on the reidentification of the vehicle trajectory, the abnormal trajectory of the vehicle can be visually displayed on the spatiotemporal map. With different judgment methods, this can quickly detect traffic incidents on the road such as vehicle speeding, vehicle parking, and traffic congestion, as shown in Fig. 8.

Judgment of vehicle speeding and parking incidents

Temporary parking or speeding on tunnels is not in compliance with laws in places such as China, and poses a serious threat to traffic safety. The speed of the vehicle can be expressed as the slope of the trajectory in the spatiotemporal map. Given a set of trajectories $Trace_{1,2,\dots,i}$, the distance traveled by vehicle i at time x_t to $x_{t+\delta t}$ is $(y_t, y_{t+\delta t})$, and the instantaneous speed of the vehicle is calculated by Eq. (13). When $Speed_{Instant}$ is higher than $Speed_{limit}$, a speeding event occurs. When $Speed_{Instant}$ is less than 5, the vehicle is considered to be temporarily stopped. In the spatiotemporal map, the speeding and parking of the vehicle can be intuitively expressed as the high and low slopes of the trajectory curve.

$$Speed_{Instant} = \frac{y_{t+\delta t} - y_t}{x_{t+\delta t} - x_t} \quad (13)$$

Fig. 6 Generation of global trajectory spatiotemporal map in multiple continuous monitoring scenes

As shown in Fig. 8a and b, the slopes of the trajectories of traffic incidents are obviously different from the normal trajectories. The dark green trajectory created by the car in Fig. 8a has a larger slope than the other two trajectories. Meanwhile, $Speed_{Instant}$ of the car is 130 km/h, which exceeds the maximum allowable speed of 120 km/h on the current road. This indicates a speeding incident. In Fig. 8b, a white vehicle parks in the inner lane, and the slope of its trajectory appears as a blue line with a slope close to 0 on the spatiotemporal map, and $Speed_{Instant}$ of the parking car is 2 km/h, and then a parking event occurs.

Judgment of traffic congestion

Based on the trajectory spatiotemporal map, road congestion can be determined. As shown in Fig. 8c, the spatiotemporal map shows that the road trajectories are very dense, and the slope tends to 0. In Eqs. (13) and (14), in the time range $(x_t, x_{t+\delta t})$ corresponding to the given distance range $(y_t, y_{t+\delta t})$, the number of trajectories in the trajectory set $Trace_{1,2,\dots,i}$ is Trace num, and the current trace density ρ_{jam} can be calculated. When ρ_{jam} is higher than $\rho_{threshold}$ and the $Speed_{Instant}$ values of vehicles are all less than 5, congestion occurs at the position of $(y_t, y_{t+\delta t})$ within the time range of $(x_t, x_{t+\delta t})$.

$$\rho_{jam} = \frac{\text{Trace num}}{x_{t+\delta t} - x_t} \quad (14)$$

Experimental results

This section tests the algorithms proposed in the third and fourth sections using an Intel i7-6800k CPU and NVIDIA GeForce RTX 3090 GPU graphics.

Vehicle object detection results

We trained the detection model from scratch and divided 80% of the dataset images into the training set. A total of 25,446 images and 51,230 different types of labeled objects participated in the training. Our test set images are 20% of the entire dataset, covering 6363 images that are not involved in training in different scenes and are labeled with ground truth. During training, some hyper-parameters are determined as follows. Our batch is set to 4, the training steps are 50,000, the learning rate is 0.001 in the first 40,000 steps, and the learning rate is changed to 0.0001 after 40,000 to ensure the convergence of the model. To enrich the diversity of objects, we performed data enhancement operations during training, which included randomly rotating the image -5° to 5° , adjusting the saturation to 1/1.5 to 1.5 times of the original image, adjusting the exposure to 1/1.5 to 1.5 times of the original image, adjusting the tone to -0.1 to 0.1 times the original image, and using the CutMix [25] method to generate a large number of enhancements to the dataset.

When the loss decreases to 0.5, it is considered that the trained model has converged, and the test set is used to measure the performance of the model. The effects of the models

Fig. 7 Global spatiotemporal map established by vehicle trajectory reidentification under multiple traffic cameras

are shown in Table 2. We use the Average Precision (AP), mean Average Precision (mAP), F1 score, and Frames Per Second (FPS) to evaluate the accuracy of the model. As shown in Eqs. (15) and (16), the precision of detection refers to the ratio of positive objects that have been correctly predicted to the total detected objects. The recall to the ratio of correctly detected objects to the total objects in the test set. TP, FN, and FP are the numbers of true positives, false negatives, and false-positives, respectively. According to Eqs. (15) and (16), the precision-recall curve of each type of object can

be drawn. AP is the integral value of the P–R curve, and mAP is the average value of all categories of APs. The F1 score is the harmonic mean of the accuracy and recall of the model, calculated as shown in Eq. (17).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (15)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (16)$$

$$\text{F1 score} = \frac{2TP}{(2TP + FN + FP)} \quad (17)$$

Fig. 8 Expression of traffic incidents on trajectory spatiotemporal map. **a** Vehicle speeding. **b** Vehicle parking. **c** Traffic congestion

We compare the original YOLOv4 network with the improved network proposed in this paper, and the results are listed in Table 2. For detection speed, our improved network is obviously faster than the original YOLOv4 network. Our improved network reached 73.34 FPS the fastest. This shows the effectiveness of using the batch encode layer. Our improved network mAP can reach 86.96%. The labeled instances are organized according to the input of the batch encode layer, which avoids the reduction of detection accuracy caused by the objects being too small and greatly improves the detection speed. In addition, using our homemade dataset significantly improves the mAP by at least 26.85% compared to using the MS COCO [41] 2017 dataset. Although there are many vehicle objects in the COCO [41] datasets, some of these vehicles are very large, and some parts of vehicles were photographed, which is far from the vehicle object posture captured in the traffic monitoring scene. Therefore, they cannot perform well on the test dataset. For the Scaled-YOLOv4 method, the YOLOv4-CSP [27] model was used for evaluation. The YOLOv4-CSP [27] model has achieved a high detection accuracy of 86.89%, and the inference speed was also 68.11 FPS, indicating that it can exert good detection performance on hardware devices. Compared with the YOLOv4-CSP [27] model, the detection method in this paper reached 86.96% in mAP, which produced competitive results. YOLOF [28] eliminated the complex and memory-consuming FPN modules and achieved 85.07% mAP using single-scale features, which matches the detection effect of our improved network. For inference speed, our detection method is approximately twice that of YOLOF [28] and is comparable to the speed of other state-of-the-art object detectors. Therefore, the improved YOLOv4 method in this paper has obvious detection speed advantages and competitive detection accuracy advantages.

Vehicle multi-object tracking results

Using the detection model to track the vehicle in the surveillance video, we made a traffic multi-object tracking dataset, which can be publicly obtained from https://drive.google.com/file/d/1itCr3McIrb08mGCZJGdF_OVV0NVN08i/view?usp=sharing. The information of the tracking dataset is listed in Table 3. The trajectory in the dataset is made according to the format of the MOT dataset [42]. The dataset covers different tunnel and highway scenes with a total of 20,000 frames, and the resolutions of the images are different. A total of 27,544 bounding boxes are manually annotated with a total of 238 trajectories.

We use a homemade tracking dataset and evaluation metrics [43,44] such as Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), recall, and Hz to compare different MOT methods. As shown in Eqs. (18) and (19), fn , fp , and IDS are the false negatives, false positives, and total number of times that a trajectory changes its matched ground truth identity, respectively. t is the frame number, g is the ground truth of detections, d_t^i is the distance between the predicted detection and i th ground truth detection in frame t , and c_t is the number of matched trajectories in frame t . MOTA shows the overall tracking accuracy, and MOTP shows the percentage alignment of the predicted bounding box and ground truth.

$$MOTA = 1 - \frac{\sum_t (fn_t + fp_t + IDS_t)}{\sum_t g_t} \quad (18)$$

$$MOTP = 1 - \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \quad (19)$$

Tracking Recall is the ratio of correctly matched trajectories to ground truth detections. IDF1 is the F1 score of the

Table 2 Comparison of state-of-the-art object detectors

Network	Dataset	AP (%)			mAP(%)	F1 score	FPS
		Car	Truck	Bus			
YOLOv4-CSP [27]	COCO [41]	43.88	50.16	51.67	48.57	0.45	68.11
	Our-day	83.41	91.87	85.39	86.89	0.85	62.24
	Our-night	63.13	85.62	79.49	76.08	0.73	63.15
	Our-tunnel	87.74	93.32	70.67	83.91	0.78	59.83
YOLOF [28]	COCO [41]	36.84	40.76	39.85	39.15	0.43	31.94
	Our-day	81.93	87.82	85.46	85.07	0.83	36.03
	Our-night	58.64	84.97	81.84	75.15	0.76	35.49
	Our-tunnel	87.35	93.54	69.07	83.32	0.78	37.67
YOLOv4 [22]	COCO [41]	43.37	49.91	51.65	48.31	0.42	35.53
	Our-day	82.84	88.99	87.53	86.45	0.82	36.45
	Our-night	62.57	86.07	80.23	76.29	0.72	37.84
	Our-tunnel	86.88	93.44	71.17	83.83	0.74	37.59
YOLOv4-our improved	COCO [41]	44.55	48.34	52.43	48.44	0.43	68.25
	Our-day	83.64	90.78	86.45	86.96	0.86	71.48
	Our-night	61.42	85.86	78.59	75.29	0.75	73.34
	Our-tunnel	87.25	93.17	71.61	84.01	0.77	69.76

Table 3 Trajectory information of homemade MOT dataset

Video name	Scene	Resolution	Number of annotated boxes	Number of traces
MOT01	Highway	1280*720	6787	64
MOT02	Highway	1280*720	5673	57
MOT03	Tunnel	1920*1080	13,591	87
MOT04	Tunnel	1920*1080	1493	30

predicted identities. Hz is the calculation speed in frames per second. Higher values of these indicators are preferred. In addition, IDSW is the identity switch, and lower values are preferred.

Table 4 shows a comparison of different tracking methods based on the traffic multi-object tracking dataset from the monitoring perspective and gives the average indicators of the comparison algorithms. Compared with the other methods, our MOTA reached 89.85, and MOTP was 4.7 higher than Deep-SORT. The tracking recall of our method reached 93.73, achieving a high correlation accuracy of vehicles. MOTDT [34] dealt with the unreliability of the online tracking problem, selected accurate candidates, and achieved a good tracking effect. In terms of tracking speed, JDE [35] improves the detection speed compared to MOTDT, but it still needs to use the Kalman filter and Hungarian algorithm, which increases the number of processing steps. Our method adopts the strategy of key frame and speed prediction, which greatly improves the tracking speed, reduces the amount of calculation, and is beneficial to the online tracking method.

Since CenterTrack [36] only involves the object association of adjacent frames, there are more id switches, and shorter trajectories often appear. In terms of traffic

monitoring scenes, the camera covers a long distance, and CenterTrack [36] cannot stably form long trajectories, but it still obtains a high MOTA of 90.45. For TraDes [37], detection and tracking tasks are combined to infer the offset of object tracking and improved object detection results, which is effective for object matching and reduced ID switching. However, the MOT algorithm in this paper does not use tracking to correct detection, but our method can still obtain 93.03 MOTA that competes with TraDes [37]. SiamMOT [38] uses the twin tracker to model the object motion, finds the matched object detection boxes in a large context area, and obtains a stable vehicle tracking effect from a monitoring perspective. SiamMOT [38] obtained a tracking recall of 93.82, close to our tracking method, which indicates the effectiveness of the key frame tracking algorithm in this paper. In terms of tracking processing speed, since CenterTrack [36], TraDeS [37], and SiamMOT [38] all need to use adjacent frame information, and number calculation tasks increases. However, our tracking algorithm only uses key frames for vehicle tracking, so our tracking algorithm had a higher tracking speed (44.07 FPS).

The above results show that our tracking method is effective for vehicle tracking in traffic scenes that require real-time

Table 4 Multi-object tracking results

Method		MOTA (↑)	MOTP (↑)	Tracking Recall (↑)	IDF1 (↑)	IDSW(↓)	HZ (↑)
Deep-SORT [33]		82.44	78.67	90.71	92.43	91	25.37
MOTDT [34]		83.34	80.16	88.59	92.75	85	28.41
JDE [35]		89.97	81.36	92.56	94.82	72	30.24
CenterTrack [36]		88.45	82.62	92.68	93.51	98	27.71
TraDeS [37]		89.13	84.26	90.74	92.38	79	25.86
SiamMOT [38]		88.49	81.19	93.82	94.11	83	21.36
Our method	MOT01	92.43	82.75	95.24	96.09	2	44.64
	MOT02	91.27	84.63	94.42	95.04	17	42.45
	MOT03	93.03	84.13	96.18	96.10	44	47.23
	MOT04	82.65	81.96	89.08	90.10	11	41.97
	Overall	89.85	83.37	93.73	94.33	74	44.07

Symbol ↑ indicates higher values are preferred, and ↓ implies lower values are expected

processing and provides correct data for drawing the trajectory spatiotemporal map.

Traffic incident detection results

To detect and analyze traffic incidents on the spatiotemporal map, we selected several representative traffic incident videos. The detailed information is shown in Table 5. These videos cover different traffic incidents and have a video resolution of 1280*720. Some incidents are concentrated at the far end of the surveillance camera, which makes the judgment of traffic incidents challenging. We also conducted experiments on the UCSD traffic video dataset. The UCSD dataset includes 20-min surveillance videos of a highway in Seattle, Washington, USA, with a resolution of 320*240. The dataset divides traffic congestion into three categories (congestion, low speed, and normal speed) and considers rainy and sunny conditions.

In the experiment, we set incident detection every 5000 frames. The road limit speed $Speed_{limit}$ is set to 90 km/h, and the congestion density threshold $\rho_{threshold}$ is set to 0.002 per/frame. Since the camera needs to be calibrated to obtain the world coordinate position of the vehicle, the video provided in Table 5 has a known height and other parameters, and the camera is calibrated. The University of California, San Diego (UCSD) dataset [45] does not give the camera parameters. According to the shooting scene and prior knowledge, we estimate that the height of the camera is 12 m, the length of the lane line is 6 m, and the width of the lane is 3.75 m. The camera is calibrated according to the above parameters. We use commonly used indicators in AID research [46] to evaluate the performance of traffic incident detection.

1. Detection Rate (DR). DR is the ratio of the number of detected traffic incidents to the total number of incidents in the video and is defined as shown in Eq. (20).

$$DR = \frac{\text{Total number of detected traffic incidents}}{\text{Total number of incidents}} \times 100\% \quad (20)$$

2. False Alarm Rate (FAR). FAR is the ratio of the number of false alarm detections to the total number of nonincident detections obtained by the algorithm, reflecting the penalty for false detections. This is defined as shown in Eq. (21).

$$FAR = \frac{\text{Total number of false alarms}}{\text{Total number of detected non - incidents}} \times 100\% \quad (21)$$

3. Mean Time to Detect (MTTD). MTTD represents the average elapsed time from the actual start of the incident to the detection of the incident by the algorithm and is defined as shown in Eq. (22).

$$MTTD = \frac{\text{Total time elapsed between detecting incidents}}{\text{Total number of correctly detected incidents}} \times 100\% \quad (22)$$

For different traffic incidents, the AID indicators of detection are calculated according to the trajectory spatiotemporal map, as shown in Table 6. From the results, it can be found that the traffic incident can be judged correctly. the DR of traffic incident detection reaches 88.09%. Although some traffic incidents occurred far away from the camera, the FAR did not increase significantly, indicating that our AID algorithm did not cause more incidents to be falsely reported while maintaining a high detection rate. Moreover, the greater the traffic volume, the more computing resources that are consumed, leading to an increase in MTTD. For the UCSD dataset [45], due to the low resolution of the video, it is easy to make

Table 5 Video information used in traffic incident detection experiments

Video name	Traffic incident	Video resolution	Video duration (s)	Video details
Video group 1	Vehicle speeding	1280*720	1107	Morning fog, high shooting angle
Video group 2	Vehicle parking	1280*720	2278	Parking at the far end of the scene
Video group 3	Traffic congestion	1280*720	2717	Congestion with single or multiple lanes

Table 6 Traffic incident detection results

Video name		DR (%)	FAR (%)	MTTD (s)
Traffic incident videos	Video group 1	91.33	21.96	2.08
	Video group 2	87.64	34.32	2.97
	Video group 3	85.29	24.13	3.34
	Overall	88.09	26.80	2.80
UCSD [46]		73.24	46.75	5.63

mistakes in trajectory matching during detection and tracking. Meanwhile, the calibration parameters are estimated, so the actual position of the vehicle may have errors, which causes a decrease in DR. However, compared with the traditional method of judging traffic incidents based on vehicle behavior characteristics in images [13,14], the trajectory spatiotemporal map used in this article can have higher judgment robustness for incident judgment regardless of the content of traffic incidents in the image. It is possible to measure the traffic situation within a certain road section from a macro perspective and not just limited to the image characteristics of the traffic scene captured by a single camera. Therefore, the construction of the trajectory spatiotemporal map has positive significance for the monitoring and control of the continuous traffic state of long roads.

Discussion

Qualitative analysis

This section gives some qualitative analysis of vehicle detection, vehicle multi-object tracking, and traffic incident detection.

In the vehicle detection method, we stitch the images into batches for object detection and then return the batch detection results to each image. As shown in Fig. 9, the batch results of vehicle detection in the day, night, and tunnel under different traffic environments are shown.

It can be seen from Fig. 9 that although the images are organized into batches to make small objects smaller, vehicle targets can still be accurately detected when they are far from the camera, and their location and category are given. This is because we built a vehicle dataset that matches the batch input. The dataset takes into account the impact of different traffic scenes on the appearance of the vehicle and

labels many small objects. Meanwhile, YOLOv4 uses mosaic data enhancement to ensure the uniform distribution of small objects. These strategies can significantly improve the detection of vehicle objects with drastic changes in size from traffic monitoring perspectives. In the lower right corner of the third single image in Fig. 9a, when the black car is partially occluded, since the PANet module improves the feature extraction ability, the object detection method in this paper can also correctly detect the object according to the local features of the vehicle, which provides a good detection basis for key frame object tracking. As shown in Fig. 9b, vehicle detection at night is challenging. Due to the lack of light, the camera produces grayscale images. The halo caused by the headlights makes it difficult to obtain the characteristics of vehicles. Therefore, the detection result at night is not as good as during the day (Fig. 9a) or in the tunnel (Fig. 9c) with good lighting conditions. However, due to the addition of night vehicle features to the vehicle dataset, the objects are also detected as much as possible.

For multi-object tracking, Fig. 10 shows some results of the key frame tracking algorithm we designed. For tracked vehicle, the yellow number provides a reference for the lane where the vehicle is located. For example, “01” means that the vehicle is driving in direction “0” in lane “1”. Figure 10a shows an extreme situation when the vehicle does not drive in a regular straight line. The magenta line is the trajectory produced by the vehicle, and the vehicle speed is displayed in cyan blue text. Although only the key frame information is used in the tracking method and the movement of vehicles in adjacent frames is not considered, stable and long-term vehicle tracking results are still achieved. This reduces the number of calculation tasks while maintaining tracking accuracy, which is preferred in real-time vehicle tracking methods.

Figure 10b shows the tracking results produced by the algorithm in this paper when there is occlusion in the traffic

(a)

(b)

(c)

Fig. 9 The batch object detection results proposed in this article. **a–c** Are the vehicle detection results during the day, night, and tunnel, respectively. Red, blue, and green boxes indicate car, truck, and bus, respectively

Fig. 10 The results of the key frame multi-object tracking method. **a** The tracking result when the vehicle is not driving in a straight line. **b** The tracking result when the vehicle is blocked

environment. In the 425th frame, the red truck tracks normally, resulting in a purple trajectory. When the red truck in the 450th frame is blocked by the road sign, according to the trajectory association matching strategies in this paper, the vehicle speed (65 km/h, cyan blue text in the image) is used to give the predicted position of the vehicle in the 475th frame, and the predicted position is used to correlate the trajectory for the vehicle. In the 520th frame, when part of the red truck moves away from the obstruction, the vehicle can be detected again using the object detection method, and the detected bounding box is used to continue tracking the vehicle. The two cars on the left side of the red truck (vehicles labeled with magenta boxes in the 450th) also followed the same strategies and continued to track after passing the obstruction. Therefore, the object trajectory matching and correlation strategies in this paper can handle a situation in which a vehicle is blocked without interrupting the trajectory and obtain a long-term vehicle running trajectory.

For traffic incident detection, Fig. 11 shows a comparison of the spatiotemporal map of vehicle parking and when only the vehicle speed is reduced. As shown in Fig. 11a, the white car stopped in the emergency lane from the 150th frame to the 480th frame, and the trajectory in the spatiotemporal map shows a stable state with a slope close to 0 for a long time. Meanwhile, in Fig. 11b, from the 50th frame to the 350th frame, the vehicle slows down in the emergency lane, showing a fluctuation of the trajectory in the spatiotemporal map. Using the spatiotemporal map, the above two cases can be easily distinguished, avoiding erroneous results caused using only image pixel analysis.

In the same way, when different incidents such as vehicle speeding and traffic jams occur, the algorithm in this paper is also concerned with the overall trend of trajectory generation in a certain period of time and does not use the state of the vehicle at certain moments to make judgments. Therefore, using the spatiotemporal map can make the traffic incident detection results convincing and reduce false-positive judgments. This is beneficial to the correct detection of traffic incidents.

Limitations

The method of using the global spatiotemporal map for traffic incident detection proposed in this manuscript can analyze the state of vehicles on continuous road sections in time, but there are some limitations. First, the vehicle speed used in the tracking algorithm and incident detection comes from the result of camera calibration. When the internal and external parameters of the camera are not accurately obtained, the camera calibration will introduce calculation errors, which affects the conversion of pixel coordinates to world coordinates, resulting in incorrect vehicle speed.

Second, many closed circuit television systems do not synchronize time between cameras during daily use. Although the algorithm in this paper does not require strict time synchronization information, when there is a large time difference between the two cameras (for example, a time difference of more than 5 seconds), the algorithm for generating the global trajectory spatiotemporal map may not obtain the correct results.

Finally, when generating a global vehicle trajectory spatiotemporal map, only continuous road sections without ramps, such as tunnels and long downhill, are considered. Although these road sections are prone to incidents and need to be considered, the situation with ramps should also be considered more extensively to improve the spatiotemporal map generation algorithm.

Future work

Driving behavior that does not comply with traffic discipline can lead to serious traffic incidents, and these dangerous driving behaviors are random and difficult to judge. This manuscript studied instances of traffic incidents such as congestion, parking, and vehicle speeding. These traffic incidents have specific rules of occurrence and often occur in the traffic environment. They continuously interfere with the smooth operation of traffic, so this article focused on these common incidents.

In future work, we will consider complex traffic driving behaviors such as frequent lane changes, skipping driving, and other abnormal driving behaviors based on the global vehicle trajectory spatiotemporal map established in this paper. These abnormal driving behaviors may appear in the trajectory spatiotemporal map as multiple extreme changes in the slope of the same vehicle trajectory. Analyzing these extreme abnormal trajectories can enrich the instances of traffic incident detection and improve the role of intelligent traffic algorithms.

In addition, we can further analyze the relationship between vehicle trajectories based on the global vehicle trajectory spatiotemporal map established in this paper: for example, whether there are vehicles whose trajectories are too close, whether a part of the trajectories in the spatiotemporal map are too dense, or whether a lane is more favored by the driver. This provides research on the car-following relationship between vehicles, the driver's preference for driving, and the safety of vehicle operation. Based on the results of the safety analysis between vehicles, deep network modeling can be used to warn against road safety risks, predict the development trends of traffic, and provide references for road traffic control. Therefore, the follow-up research of this work aims to more deeply examine traffic monitoring video information and provide valuable traffic travel guide services.

Fig. 11 Comparison of vehicle parking and deceleration in the spatiotemporal map. **a** Vehicle parking. **b** Vehicle deceleration

Conclusion

This research proposed a method to analyze traffic incidents using spatiotemporal maps of trajectories from monitoring perspectives. According to the characteristics of different traffic scenes, we self-labeled the vehicle datasets used in the daytime, tunnel, and at night, and used the improved YOLOv4 network to batch process traffic images to quickly obtain the vehicle detection results of multiple images. According to the vehicle detection results, the tracking method of vehicle speed prediction in the key frame was adopted. This reduced the calculation overhead of tracking and obtained complete vehicle trajectories. According to the vehicle trajectories in separate scenes, we reidentified the trajectory, measured the similarity between the spatial position of the trajectory and the image ROI to correlate the same vehicle trajectory in different traffic scenes, and constructed a spatiotemporal map of the trajectory in continuous traffic scenes. Based on an analysis of the spatiotemporal map, the detection methods of traffic incidents were given.

Experiments showed that the detection and tracking method proposed in this paper can greatly increase the calculation speed while taking into account the accuracy and meeting real-time online processing requirements. The complete spatiotemporal map constructed by trajectory reidentification provides reliable and intuitive data for traffic incident analysis. This research can detect and report traffic incidents in time. In particular, this article can provide an important reference for traffic analysis of continuous road sections captured by multiple surveillance cameras, such as long tunnels, to ensure traffic safety.

Funding Funding This work was supported by the National Key Research and Development Program of China (2019YFB1600702), the National Natural Science Foundation of China (No. 6207072223), the

National Natural Science Funds for Distinguished Young Scholars of China (No. 62006026).

Data Availability Statement The vehicle tracking dataset generated in fifth section is available in the Google Drive, https://drive.google.com/file/d/1itCr3McIrb08mGCZJGdF_OVV0NVN08i/view?usp=sharing. Other datasets analysed during the current study are not publicly available due to privacy reasons. It was assured that data generated for the research project are only used for this research context.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there do not have any conflicts of interest to declare.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Wells T, Toffin E (2005) Video-based automatic incident detection on san-mateo bridge in the san francisco bay area. In: 12th World Congress on ITS, San Francisco, Citeseer
2. Williams BM, Guin A (2007) Traffic management center use of incident detection algorithms: findings of a nationwide survey. *IEEE Trans Intell Transp Syst* 8(2):351–358
3. Naphade M, Chang MC, Sharma A, Anastasiu DC, Jagarlamudi V, Chakraborty P, Huang T, Wang S, Liu MY, Chellappa R et al (2018)

- The 2018 nvidia ai city challenge. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 53–60
4. Naphade M, Tang Z, Chang MC, Anastasiu DC, Sharma A, Chellappa R, Wang S, Chakraborty P, Huang T, Hwang JN et al (2019) The 2019 ai city challenge. In: CVPR Workshops, vol 8
 5. Han J, Zhang D, Cheng G, Liu N, Xu D (2018) Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Process Mag* 35(1):84–100
 6. Ozkurt C, Camci F (2009) Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. *Math Comput Appl* 14(3):187–196
 7. Kotzenmacher J, Minge ED, Hao B (2005) Evaluation of portable non-intrusive traffic detection system. *IMSA J* 43:3
 8. Zhong M, Guoxin L (2007) Establishing and managing jurisdiction-wide traffic monitoring systems: North american experiences. *J Transp Syst Eng Inf Technol* 7(6):25–38
 9. Balcilar M, Sonmez AC (2008) Extracting vehicle density from background estimation using kalman filter. In: 2008 23rd international symposium on computer and information sciences, IEEE, pp 1–5
 10. Ren S, He K, Girshick R, Sun J (2016) Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 39(6):1137–1149
 11. Onoro-Rubio D, López-Sastre RJ (2016) Towards perspective-free object counting with deep learning. In: European conference on computer vision, Springer, pp 615–629
 12. Asmaa O, Mokhtar K, Abdelaziz O (2013) Road traffic density estimation using microscopic and macroscopic parameters. *Image Vis Comput* 31(11):887–894
 13. Lempitsky V, Zisserman A (2010) Learning to count objects in images. *Adv Neural Inf Process Syst* 23:1324–1332
 14. Gonçalves WN, Machado BB, Bruno OM (2012) Spatiotemporal gabor filters: a new method for dynamic texture recognition. [arXiv:1201.3612](https://arxiv.org/abs/1201.3612)
 15. Yuan Z, Zhou X, Yang T (2018) Hetero-convlstm: a deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & Data Mining, pp 984–992
 16. Hashemi H, Abdelghany K (2018) End-to-end deep learning methodology for real-time traffic network management. *Comput-Aided Civ Infrastruct Eng* 33(10):849–863
 17. Zhu L, Guo F, Krishnan R, Polak JW (2018) A deep learning approach for traffic incident detection in urban networks. In: 2018 21st international conference on intelligent transportation systems (ITSC), IEEE, pp 1011–1016
 18. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
 19. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
 20. Hu X, Xu X, Xiao Y, Chen H, He S, Qin J, Heng PA (2018) Sinet: a scale-insensitive convolutional neural network for fast vehicle detection. *IEEE Trans Intell Transp Syst* 20(3):1010–1019
 21. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision, Springer, pp 21–37
 22. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: optimal speed and accuracy of object detection. [arXiv:2004.10934](https://arxiv.org/abs/2004.10934)
 23. Purkait P, Zhao C, Zach C (2017) Spp-net: deep absolute pose regression with synthetic views. [arXiv:1712.03452](https://arxiv.org/abs/1712.03452)
 24. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8759–8768
 25. Yun S, Han D, Oh SJ, Chun S, Choe J, Yoo Y (2019) Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 6023–6032
 26. Chen Y, Zhang P, Li Z, Li Y, Zhang X, Meng G, Xiang S, Sun J, Jia J (2020) Stitcher: feedback-driven data provider for object detection. [arXiv-2004](https://arxiv.org/abs/2004)
 27. Wang CY, Bochkovskiy A, Liao HYM (2021) Scaled-yolov4: scaling cross stage partial network. In: Proceedings of the IEEE/cvf conference on computer vision and pattern recognition, pp 13029–13038
 28. Chen Q, Wang Y, Yang T, Zhang X, Cheng J, Sun J (2021) You only look one-level feature. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13039–13048
 29. Xie E, Ding J, Wang W, Zhan X, Xu H, Li Z, Luo P (2021) Detco: unsupervised contrastive learning for object detection. [arXiv:2102.04803](https://arxiv.org/abs/2102.04803)
 30. Reid D (1979) An algorithm for tracking multiple targets. *IEEE Trans Autom Control* 24(6):843–854
 31. Fortmann T, Bar-Shalom Y, Scheffe M (1983) Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J Oceanic Eng* 8(3):173–184
 32. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B (2016) Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP), IEEE, pp 3464–3468
 33. Wojke N, Bewley A, Paulus D (2017) Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP), IEEE, pp 3645–3649
 34. Chen L, Ai H, Zhuang Z, Shang C (2018) Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE international conference on multimedia and expo (ICME), IEEE, pp 1–6
 35. Wang Z, Zheng L, Liu Y, Li Y, Wang S (2020) Towards real-time multi-object tracking. In: Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, Springer, pp 107–122
 36. Zhou X, Koltun V, Krähenbühl P (2020) Tracking objects as points. In: European conference on computer vision, Springer, pp 474–490
 37. Wu J, Cao J, Song L, Wang Y, Yang M, Yuan J (2021) Track to detect and segment: an online multi-object tracker. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12352–12361
 38. Shuai B, Berneshawi A, Li X, Modolo D, Tighe J (2021) Siammot: Siamese multi-object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12372–12382
 39. Liu S, Liu D, Srivastava G, Połap D, Woźniak M (2021) Overview and methods of correlation filter algorithms in object tracking. *Compl Intell Syst* 7(4):1895–1917
 40. Lucas BD, Kanade T et al (1981) An iterative image registration technique with an application to stereo vision. In: Proceedings of the seventh international joint conference on artificial intelligence, Vancouver, British Columbia, pp 674–679
 41. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision, Springer, pp 740–755
 42. Milan A, Leal-Taixé L, Reid I, Roth S, Schindler K (2016) Mot16: a benchmark for multi-object tracking. [arXiv:1603.00831](https://arxiv.org/abs/1603.00831)
 43. Bernardin K, Stiefelwagen R (2008) Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP J Image Video Process* 2008:1–10
 44. Li Y, Huang C, Nevatia R (2009) Learning to associate: Hybrid-boosted multi-target tracker for crowded scene. In: 2009 IEEE conference on computer vision and pattern recognition, IEEE, pp 2953–2960

45. Chan AB, Vasconcelos N (2005) Classification and retrieval of traffic video using auto-regressive stochastic processes. In: IEEE proceedings intelligent vehicles symposium, IEEE, pp 771–776
46. Srinivasan D, Jin X, Cheu RL (2005) Adaptive neural network models for automatic incident detection on freeways. *Neurocomputing* 64:473–496

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.