



OPEN ACCESS

EDITED BY

Guanqiu Qi,
Buffalo State College, United States

REVIEWED BY

Yunze Wang,
Shijiazhuang Tiedao University, China
Zhe Li,
Hunan University, China
Ye Li,
Central South University, China

*CORRESPONDENCE

Yanqiu Bi,
✉ biyanqiu@cqjtu.edu.cn

RECEIVED 16 January 2024

ACCEPTED 07 February 2024

PUBLISHED 19 February 2024

CITATION

Luo Z, Bi Y, Yang X, Li Y, Yu S, Wu M and Ye Q (2024), Enhanced YOLOv5s + DeepSORT method for highway vehicle speed detection and multi-sensor verification. *Front. Phys.* 12:1371320. doi: 10.3389/fphy.2024.1371320

COPYRIGHT

© 2024 Luo, Bi, Yang, Li, Yu, Wu and Ye. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Enhanced YOLOv5s + DeepSORT method for highway vehicle speed detection and multi-sensor verification

Zhongbin Luo^{1,2}, Yanqiu Bi^{3,4*}, Xun Yang^{1,2}, Yong Li^{5,6}, Shanchuan Yu^{1,2}, Mengjun Wu^{1,2} and Qing Ye^{1,2}

¹China Merchants Chongqing Communications Research and Design Institute Co., Ltd., Chongqing, China, ²Research and Development Center of Transport Industry of Self-Driving Technology, Chongqing, China, ³National and Local Joint Engineering Research Center of Transportation Civil Engineering Materials, Chongqing Jiaotong University, Chongqing, China, ⁴School of Civil Engineering, Chongqing Jiaotong University, Chongqing, Shandong, China, ⁵College of Computer Science, Chongqing University, Chongqing, China, ⁶Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing University, Chongqing, China

Addressing the need for vehicle speed measurement in traffic surveillance, this study introduces an enhanced scheme combining YOLOv5s detection with Deep SORT tracking. Tailored to the characteristics of highway traffic and vehicle features, the dataset data augmentation process was initially optimized. To improve the detector's recognition capabilities, the Swin Transformer Block module was incorporated, enhancing the model's ability to capture local regions of interest. *CIoU* loss was employed as the loss function for the vehicle detection network, accelerating model convergence and achieving higher regression accuracy. The Mish activation function was utilized to reduce computational overhead and enhance convergence speed. The structure of the Deep SORT appearance feature extraction network was modified, and it was retrained on a vehicle re-identification dataset to mitigate identity switches due to obstructions. Subsequently, using known references in the image such as lane markers and contour labels, the transformation from image pixel coordinates to actual coordinates was accomplished. Finally, vehicle speed was measured by computing the average of instantaneous speeds across multiple frames. Through radar and video Multi-Sensor Verification, the experimental results show that the mean Average Precision (mAP) for target detection consistently exceeds 90%. The effective measurement distance for speed measurement is around 140 m, with the absolute speed error generally within 1–8 km/h, meeting the accuracy requirements for speed measurement. The proposed model is reliable and fully applicable to highway scenarios.

KEYWORDS

YOLOv5S, Deep SORT, swin transformer, vehicle speed, traffic monitoring

1 Introduction

Intelligent Transportation Systems (ITS) have been widely applied to practical traffic scenarios such as highways, urban roads, tunnels, and bridges. This integration owes much to the convergence of various technologies, including pattern recognition, video image processing, and network communication [1, 2]. Vehicle speed is a crucial parameter that

directly reflects the state of traffic [3, 4]. Meanwhile, in highly complex traffic monitoring scenarios and under special weather conditions, intelligent transportation monitoring systems face numerous significant challenges. In addressing the issue of vehicle speeding, the measurement of vehicle speed can provide vital data for traffic management authorities. Accurate measurement of vehicle target speed is one of the challenges faced by traffic monitoring systems.

Traditional vehicle speed detection primarily utilizes inductive loop detection, laser detection, and radar detection. These methods are well-developed and commonly used in traffic systems. However, traditional detection methods have the following disadvantages: (1) the required equipment is expensive; (2) the equipment is installed under the road surface, leading to high subsequent maintenance costs and maintenance not only affects traffic but also damages road structure. Video-based vehicle speed detection leverages numerous traffic video monitoring devices, significantly overcoming the high costs and difficult maintenance issues associated with traditional speed detection methods. The vehicle speed detection system can be categorized into two types: one type focuses on accurate speed monitoring systems (such as speed camera applications) [5, 6], and the other type, though less precise, can be used to estimate traffic speed (such as traffic camera application scenarios) [7, 8]. This classification system takes into account the intrinsic parameters of the camera (such as sensor size and resolution, focal length), as well as extrinsic parameters (such as the camera's position relative to the road surface, drone-based cameras, etc.), and the number of cameras (monocular, stereo, or multiple cameras).

Through these parameters, the actual scene on the image plane can represent one or multiple lanes, as well as the relative position of vehicles to the camera, ultimately yielding one of the most critical variables: the ratio of pixels to road segment length, i.e., the road length each pixel represents. Due to the perspective projection model, this ratio is directly proportional to the square of the camera's distance, implying that measurements over long distances have poor accuracy. Accurate estimation of the camera's intrinsic and extrinsic parameters is required to provide measurements in the actual coordinate system. The most common approach is soft calibration, which involves calibrating intrinsic parameters in a verification laboratory or using sensor and lens characteristics, and estimating the rigid transformation between the camera and the road surface using manual [9, 10] or automatic [11] methods.

Hard calibration involves estimating both the intrinsic and extrinsic parameters of the camera, which can be done either manually [12] or automatically [13–15]. In certain limited scenarios, some details of camera calibration may be overlooked, such as the exact position of the camera, anchoring systems, gantries. Since cameras are mostly static (except for drone cameras), vehicle detection is most often addressed by modeling the background [16–18]. Other methods are feature-based, such as detecting vehicle license plates [19, 20] or other characteristics [21–23].

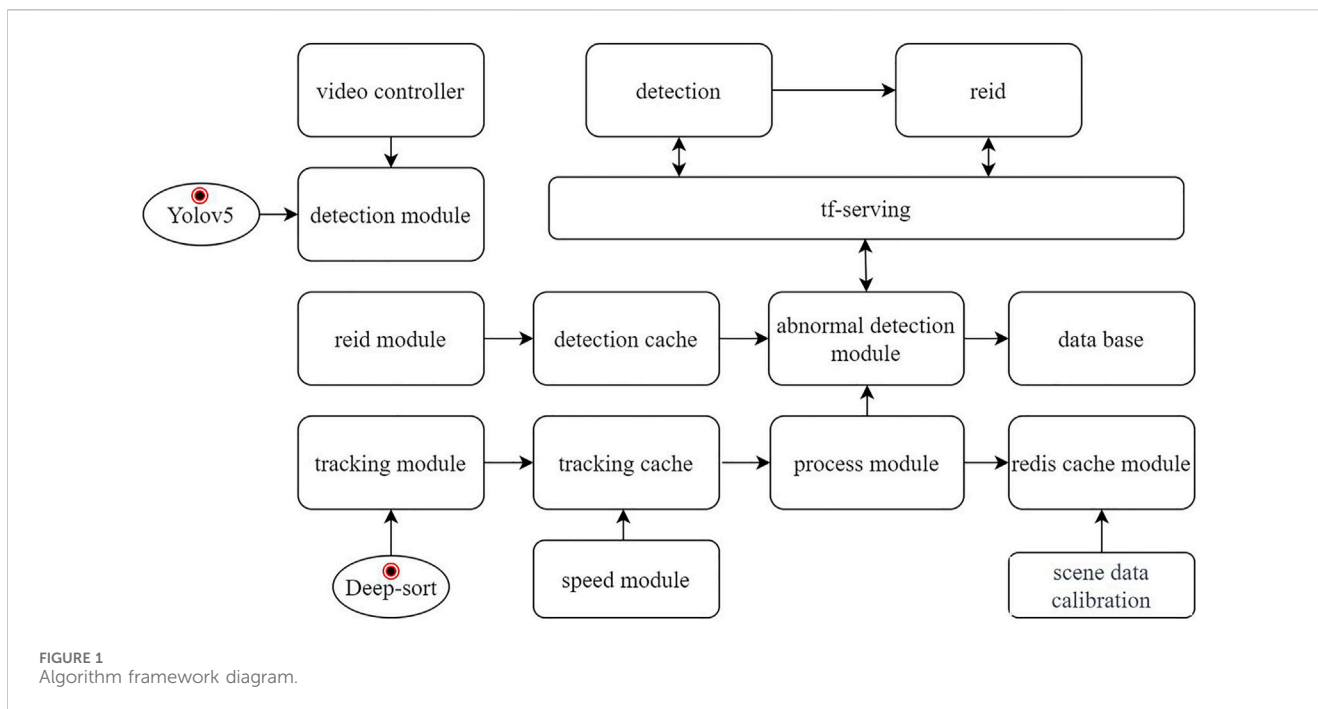
Recently, learning-based approaches have become increasingly popular for recognizing vehicles in images [24, 25]. The ability to track vehicles with smooth and stable trajectories is a key issue in handling vehicle speed detection. Vehicle tracking can be divided into three different categories: The first category is feature-based [26–28], where tracking originates from a set of features of the

vehicle (such as optical flow). The second category focuses on tracking the centroid of a vehicle's blob or bounding box [29, 30]. The third category concentrates on tracking the entire vehicle [31, 32] or its specific parts (such as the license plate [33, 34]).

The prerequisite for speed measurement is the effective assessment of distance. In monocular vision systems, the estimation of vehicle distance typically relies on specific constraints and methods. These include: (1) Flat road assumption and homography-based methods, which assume that the road is flat and apply a mathematical transformation known as homography [35, 36], helping in mapping the view of a scene from one perspective to another, which is crucial for estimating distances in 2D images; (2) Detection of lines and specific areas [37, 38]. By detecting lines and specific areas, designed detection lines and areas can be overlaid on the real-world view, providing a reference scale for measuring distances; (3) Use of prior knowledge about object dimensions, utilizing the known dimensions of certain objects to estimate distances. For instance, knowing the standard sizes of license plates ([39, 40]) or the average dimensions of vehicles [41] can assist in calibrating distance measurements. However, these monocular methods have limitations, which are addressed in stereo vision systems. In stereo vision systems [42], two cameras are used to capture the same scene from slightly different angles, similar to human binocular vision. This setup allows for more accurate depth perception and distance estimation, as it mimics the way.

Currently, speed detection is primarily divided into macroscopic traffic flow speed and individual vehicle speed. Macroscopic traffic flow speed detection is based on a specific road section, using the length of the section and travel time to estimate the average speed of the segment [43, 44]. Individual vehicle speed detection focuses on the micro-level speed of the vehicle itself, presenting greater technical challenges. This process requires prior knowledge of the camera's frame rate or accurate timestamps for each image to calculate the time between measurements. Utilizing consecutive or non-consecutive [45] images to estimate speed is a key factor impacting accuracy. In summary, whether in traffic flow speed or individual vehicle speed detection, factors such as the method of image capture (continuous or non-continuous), frame rate, timestamp accuracy, and the integration of various measurement data need to be carefully considered. The selection method and precision of these factors directly affect the accuracy of speed estimation.

In summary, vision-based vehicle speed detection involves the entire process of camera calibration, distance estimation, and speed estimation. However, the calibration process for monocular vision cameras is complex, the accuracy of distance estimation is relatively poor, and the precision of individual vehicle speed estimation needs improvement. Currently, there are few instances of rapidly detecting and stably tracking vehicle instantaneous speeds solely through video recognition technology, which limits the broader application of video recognition technologies in the field of traffic safety. Therefore, this study introduces an enhanced scheme that combines YOLOv5s detection with Deep SORT tracking, targeting the need for vehicle speed measurement in traffic monitoring. The dataset data expansion process is preliminarily optimized based on the characteristics of highway traffic and vehicle features. The Swin Transformer Block module is introduced to improve the detector's



recognition capabilities and enhance the model's ability to capture areas of interest. The $CIoU$ loss is employed as the loss function for the vehicle detection network to accelerate model convergence and achieve higher regression precision. The Mish activation function is used to reduce computational costs and improve convergence speed. Modifications are made to the structure of the Deep SORT appearance feature extraction network, and it is retrained on the vehicle re-identification dataset to mitigate identity switches caused by obstacles. Subsequently, known references in the image, such as lane markings and contour labels, are used to complete the conversion from image pixel coordinates to actual coordinates through maximum likelihood estimation, maximum posterior estimation, and non-linear least squares methods. Finally, vehicle speed is measured by calculating the average of instantaneous speeds over multiple frames. The algorithm can detect and track vehicle targets without prior camera parameters and calibration, extract known reference information such as lane lines and contour labels, and automatically convert pixel coordinates to actual coordinates in traffic monitoring scenes, as well as automatically measure vehicle speeds, the algorithm framework as shown in Figure 1. Accurate estimation of vehicle speed can support the detection of traffic accidents and incidents, offering scientific technical means for active safety management in intelligent transportation systems.

2 Improved YOLOv5s + DeepSORT algorithm for highway vehicle detection and tracking

2.1 Construction of vehicle target dataset

2.1.1 Characteristics of highway traffic scenarios

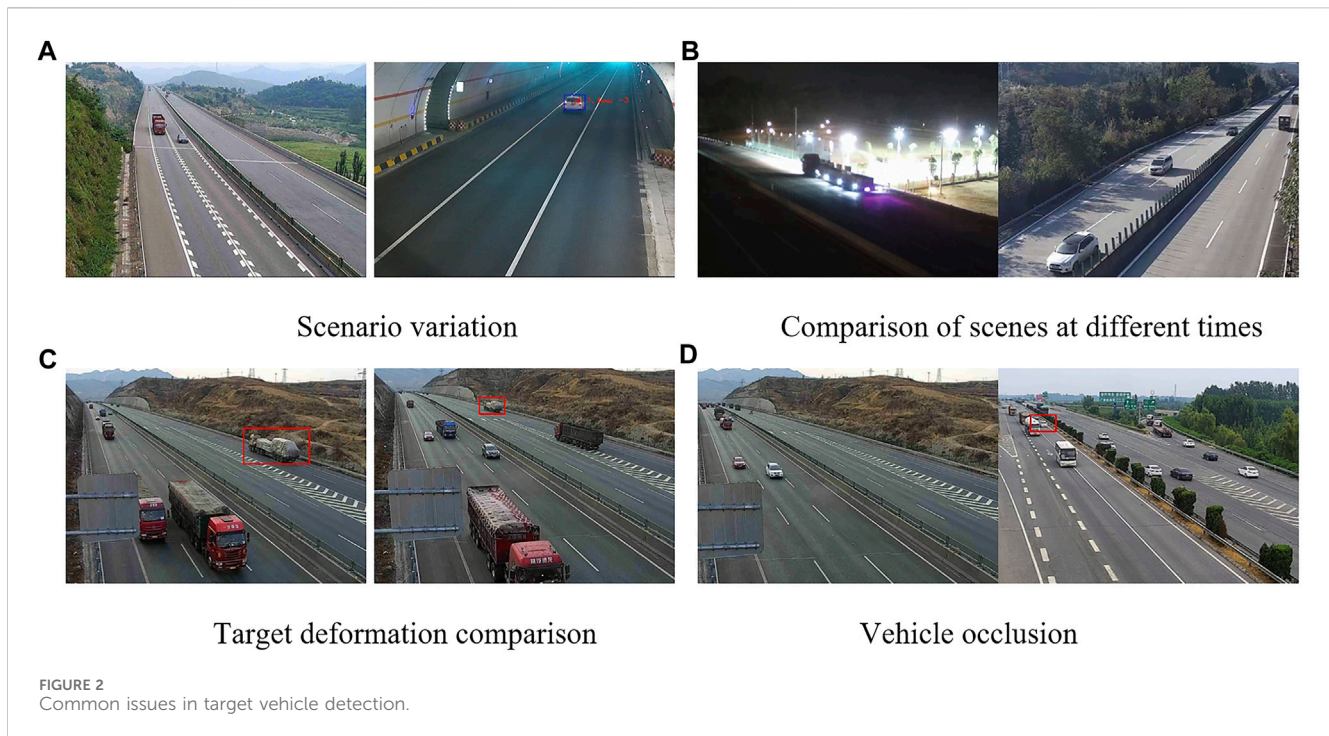
There are typically four categories of common highway traffic scenarios, as shown in Figure 2.

- Scene variations, as the setup of traffic monitoring varies, so do the monitoring angles and heights. For instance, the monitoring angle and scene characteristics inside a tunnel differ greatly from those on a highway, leading to significantly reduced detection accuracy and numerous false detections of vehicle targets, as shown in Figure 2A.
- The same scene at different times also exhibits significant differences. With changes in time, the brightness and visibility of scene images vary. The characteristics of vehicle targets at night are particularly difficult to capture due to the substantial interference from vehicle lights at night, making it hard to accurately obtain the body contours of target vehicles. If the dataset does not include such special night scene data, the detection results are not ideal Figure 2B.
- Vehicle targets at different positions in the image will have obvious deformation. The same vehicle target will undergo significant size deformation from distant to closer positions in the image, affecting the detection accuracy of small targets. The red boxes in Figure 2C indicate significant deformations of the same vehicle target at different locations.
- On actual roads, there is a widespread occurrence of vehicle occlusion, which can lead to multiple targets being detected as one, resulting in missed and false detections. The red boxes in Figure 2D represent situations where vehicles are obstructing each other.

The existence of these four types of issues makes large public datasets such as COCO and VOC unsuitable for the perspectives captured by highway cameras, leading to a large number of false positives and missed detections of target vehicles.

2.1.2 Data preparation

Given the relatively uniform types of motor vehicles in highway scenarios, vehicles are generally classified into three



categories: Car, Bus, and Truck. Car mainly refer to passenger vehicles with seating for fewer than seven people; Bus mainly include commercial buses, public transport buses, etc.; Truck primarily refer to small, medium, and large trucks, trailers, and various types of special-purpose vehicles as shown in Table 1. By collecting datasets from different scenes on highways and manually labeling them using the labelImg tool, a dataset in YOLO format was ultimately created.

The specific process includes: (1) Data Collection: Collect representative image data covering various scenes and angles of target categories. (2) Data Division: Divide the dataset into training, validation, and test sets, typically in a certain ratio, to ensure the independence and generalizability of the data. (3) Bounding Box Annotation: Annotate each target object with a bounding box, usually represented by a rectangle, including the coordinates of the top-left and bottom-right corners. Category Labeling: Assign corresponding category labels to each target object, identifying the category to which the object belongs. During dataset annotation, rectangular bounding boxes encompassing the entire vehicle are marked, with each side fitting closely to the vehicle. Annotation is not performed when the occlusion exceeds 50%, the vehicle type is indistinguishable, or the size is below 10×10 pixels. Furthermore, in cases where vehicles are truncated, the truncation is not considered to affect the overall annotation. Trucks used for transportation are uniformly annotated, without separately marking the vehicles on them.

2.1.3 Data augmentation

To enhance the accuracy and generalization capability of model training, data augmentation techniques are employed, tailored to the characteristics of highway traffic environments and vehicle features. These techniques include Mosaic, Random_perspective, Mixup, HSV, Flipud, Fliplr, as shown in Figure 3.

2.2 Optimization of object detection network

In response to the identified issues with YOLOv5 in highway vehicle detection, the following optimizations were made to enhance the accuracy of vehicle detection: (1) Incorporating the Swin Transformer Block module to improve the model's ability to capture information from local areas of interest; (2) Utilizing *CIoU* loss as the loss function for the vehicle detection network to accelerate model convergence and achieve higher regression accuracy; (3) Adopting the Mish activation function to reduce computational overhead and increase convergence speed.

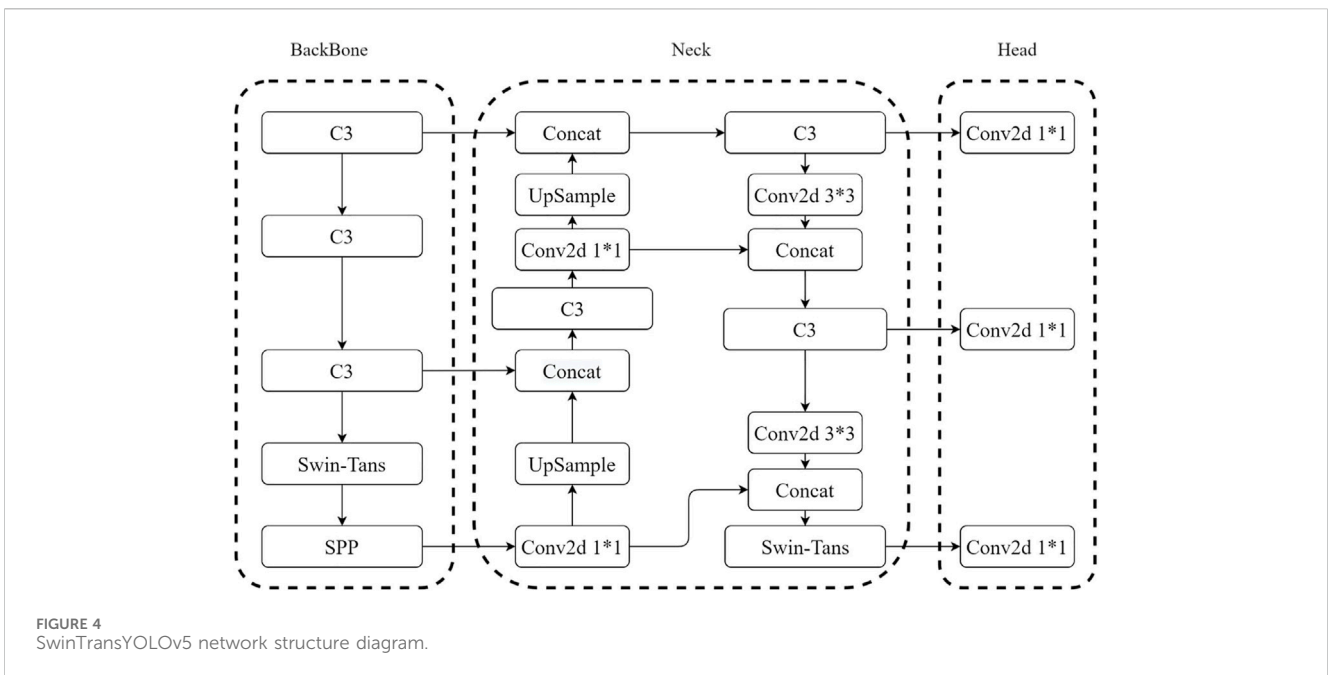
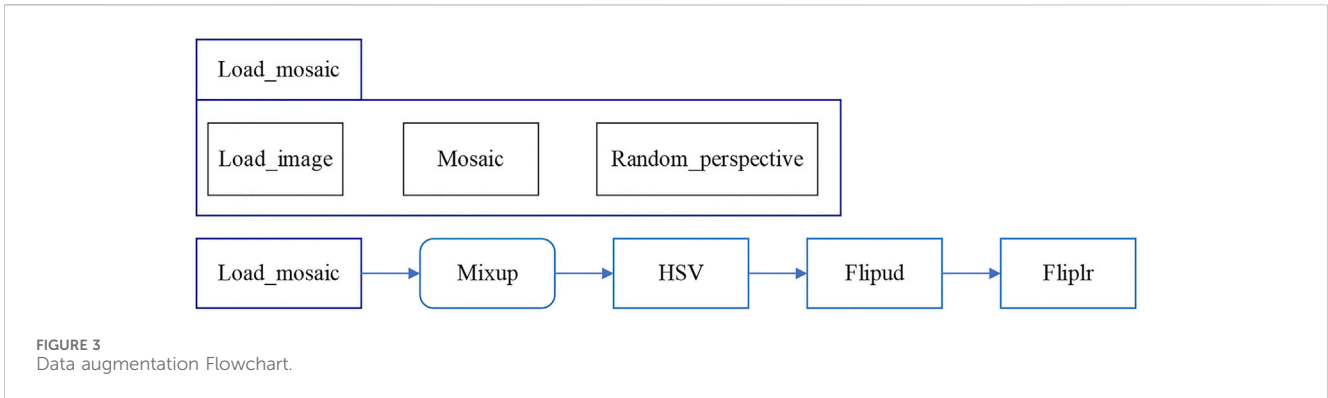
2.2.1 Introduction of swin transformer block

To address the shortcomings of traditional YOLOv5 in traffic object detection, the Swin Transformer Block module is introduced for optimization.

The Swin Transformer network [46], proposed in 2021, is a Transformer network enhanced with a local self-attention mechanism. It has stronger dynamic computation capabilities compared to convolutional neural networks, with enhanced modeling capacity, and can adaptively compute both local and global pixel relationships, making it highly valuable for widespread use.

The core modules of the Transformer Block overall architecture are the Window-based Multi-Head Self-Attention layer (W-MSA) and the Shifted Window-based Multi-Head Self-Attention layer (SW-MSA). By restricting attention computation within a window, the network not only introduces the locality of convolution operations but also saves computational resources, resulting in good performance.

This article proposes the integration of the Swin Transformer Block structure into the backbone feature extraction network and



neck feature fusion, utilizing the efficient self-attention mechanism module to fully explore the potential of feature representation. The improved YOLOv5 network incorporating the Swin Transformer Block module is shown in Figure 4, named SwinTransYOLOv5 network.

2.2.2 Improvement of loss function

YOLOv5s employs *GIoU* loss as the bounding box regression loss function to evaluate the distance between the predicted bounding box (PB) and the ground truth bounding box (GT), as shown in Eq. 1.

$$\begin{cases} GIoU = IoU - \frac{A^c - U}{A^c} \\ L_{GIoU} = 1 - GIoU \end{cases} \quad (1)$$




In the formula, *IoU* represents the intersection over union of PB and GT, A^c is the area of the smallest rectangular box containing both PB and GT, U is the union of PB and GT, and L_{GIoU} is the *GIoU* loss. The advantage of *GIoU* loss is its scale invariance, meaning the

similarity between PB and GT is independent of their spatial scale. The problem with *GIoU* Loss is that when either PB or GT completely encompasses the other, *GIoU* Loss degenerates entirely into *IoU* loss. Because it heavily relies on the *IoU* term, this results in slow convergence during actual training and lower accuracy of the predicted bounding boxes. To address these issues, *CIoU* loss also considers the overlapping area of PB and GT, the distance between their centroids, and their aspect ratios, as shown in Eq. 2.

$$\begin{cases} CIoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - av \\ L_{CIoU} = 1 - CIoU \end{cases} \quad (2)$$

In the formula, b and b^{gt} represent the centroids of PB and GT, $\rho^2(\cdot)$ denotes the Euclidean distance, c is the length of the shortest diagonal of the smallest enclosing box of PB and GT, a represents a positive balance parameter, and v indicates the consistency of the aspect ratio of PB and GT. The definitions of a and v are as follows in Eq. 3.

TABLE 1 Dataset categorization.

Type	Example
Car	
Bus	
Truck	

$$\begin{cases} v = \frac{4}{\pi^2} \left(\arctan \frac{\omega^{gt}}{h^{gt}} - \arctan \frac{\omega}{h} \right) \\ a = \frac{v}{(1 - IoU) + v} \end{cases} \quad (3)$$

In the formula, ω^{gt} , h^{gt} and ω , h respectively represent the width and height of GT and PB.

Compared to the $GIoU$ loss used in YOLOv5s, $CIoU$ loss incorporates penalty terms for the distance between the centers of PB and GT, as well as their aspect ratios in the loss function. This ensures faster convergence of the predicted bounding boxes during training and yields higher regression localization accuracy. In This article, $CIoU$ loss is adopted as the loss function for the vehicle detection network.

2.2.3 Activation function

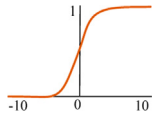
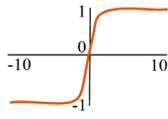
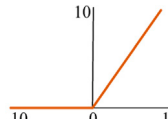
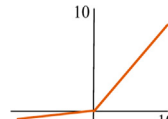
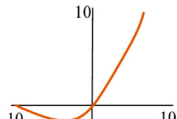
Changing the activation function can significantly enhance recognition performance. Activation functions are categorized into saturated and non-saturated types. The primary advantages of using non-saturated activation functions are twofold [47]: firstly,

they effectively address the vanishing gradient problem, which becomes more severe with saturated activation functions; secondly, they can accelerate the convergence speed. After comparing the pros, cons, and characteristics of various activation functions without significantly increasing computational load, as shown in Table 2, the Leaky ReLU activation function in YOLOv5 was replaced with the Mish activation function.

2.3 Optimization of deep SORT for vehicle tracking

The multi-object online tracking algorithm SORT [48] (Simple Online and Realtime Tracking) utilizes Kalman filtering and Hungarian matching, using the IoU between tracking and detection results as the cost matrix, to implement a simple, efficient, and practical tracking paradigm. However, the SORT algorithm's limitation lies in its association metric being effective only when the uncertainty in state estimation is low, leading to

TABLE 2 Comparison of common activation functions.

	Sigmoid	tanh	ReLU	Leaky ReLU	Mish
Function graphs					
Function Formula	$\delta(x) = \frac{1}{1+e^{-x}}$	$\tanh(x)$	$\max(0, x)$	$\max(0.1x, x)$	$x^* \tanh(\text{softplus}(x))$
Advantages	Can restrict the output to be between (0, 1), facilitating the completion of classification tasks	① Can restrict the output to be between (-1, 1), facilitating the completion of classification tasks	Linear: Saves computational resources and shortens convergence time	① Linear	① Linear
		② Zero-Centered		② Gradient non-saturation, no neuron death	② Gradient non-saturation, no neuron death
					③ The network's convergence is the best among the five activation functions
Disadvantages	① The output is not zero-centered, leading to a zigzag pattern in gradient descent	② Gradient saturation, Gradient vanishing	Neuron Death: The left side of the ReLU function is completely flat. When the neuron's z-value is negative, the output α is 0, and the gradient is also 0, making it impossible to alter the weight value w through the gradient, leaving w unchanged	The network's convergence is not advantageous compared to the latest networks	Relatively higher computational cost
	② Gradient saturation, Gradient vanishing	③ Non-linear			
	③ Non-linear, involves exponential operations, consuming more resources during computation				

numerous identity switches and tracking failures when the target is occluded. To address this issue, Deep SORT [49] combines both motion and appearance information of the target as the association metric, improving tracking failures caused by the target's disappearance and reappearance.

2.3.1 Tracking processing and state estimation

Deep SORT uses an 8-dimensional state space $(u, v, \gamma, h, x, y, \gamma, h)$ to describe the target's state and motion information in the image coordinate system. u and v represent the center coordinates of the target detection box, γ and h respectively represent the aspect ratio and height of the detection box, and (x, y, γ, h) represent the relative velocity of the previous four parameters in the image coordinates. The algorithm employs a standard Kalman filter with a constant velocity model and a linear observation model, using the detection box parameters (u, v, γ, h) as direct observations of the object state. By combining motion and appearance information, the Hungarian algorithm is used to match predicted and tracked boxes, and cascaded matching is integrated to enhance accuracy.

(1) Mahalanobis Distance

The Mahalanobis distance is used to evaluate the predicted Kalman state and the new state, as shown in Eq. 4.

$$d^{(1)}(i, j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \tag{4}$$

$d^{(1)}(i, j)$ represents the motion matching degree between the j detection and the i trajectory, where S_i is the covariance matrix of the observation space at the current moment predicted by the Kalman filter for the trajectory, y_i is the predicted observation of the trajectory at the current moment, and d_j is the state of the j detection.

Considering the continuity of motion, detections are filtered using this Mahalanobis distance, with the 0.95 quantile of the chi-square distribution as the threshold value, defining a threshold function, as shown in Eq. 5.

$$b_{ij}^{(1)} = 1 [d^{(1)}(i, j) \leq t^{(1)}] \tag{5}$$

(2) Appearance features

While Mahalanobis distance is a good measure of association when the target's motion uncertainty is low, it becomes ineffective in practical situations like camera movement, leading to a large number of mismatches. Therefore, we integrate a second metric. For each BBox detection, we compute an appearance feature descriptor. We create a gallery to store the descriptors of the latest 100 trajectories and then use the minimum cosine distance between the i and j trajectories as the second measure, as shown in Eq. 6.

$$d^{(2)}(i, j) = \min\{1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \in \mathcal{R}_i\} \tag{6}$$

Can be represented using a threshold function, as shown in Eq. 7.

TABLE 3 Adjusted reconstruction network.

Network layer	Convolutional kernel parameters	Output size
Conv 1	3 × 3/1	32 × 128×128
Conv 2	3 × 3/1	32 × 128×128
Max Pool 3	3 × 3/2	32 × 64×64
Residual 4	3 × 3/1	32 × 64×64
Residual 5	3 × 3/1	32 × 64×64
Residual 6	3 × 3/2	64 × 32×32
Residual 7	3 × 3/1	64 × 32×32
Residual 8	3 × 3/2	128 × 16×16
Residual 9	3 × 3/1	128 × 16×16
Dense 10	-	128
Batch and ℓ ₂ Norm	-	128

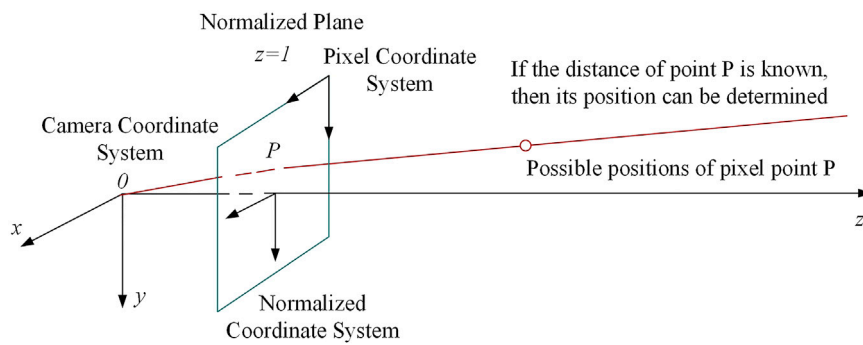


FIGURE 5 Pixel coordinate conversion diagram.

$$b_{ij}^{(2)} = 1 [d^{(2)}(i, j) \leq t^{(2)}] \tag{7}$$

Mahalanobis distance can provide reliable target location information in short-term predictions, and the cosine similarity of appearance features can recover the target ID when the target is occluded and reappears. To make the advantages of both measures complementary, a linear weighting approach is used for their combination, as shown in Eqs 8, 9.

$$c_{ij} = \lambda d^{(1)}(i, j) + (1 - \lambda) d^{(2)}(i, j) \tag{8}$$

$$b_{i,j} = \prod_{m=1}^2 b_{i,j}^{(m)} \tag{9}$$

In summary, distance measurement is effective for short-term prediction and matching, while appearance information is more effective for matching long-lost trajectories. The choice of hyperparameters depends on the specific dataset. For datasets with significant camera movement, the degree of motion matching is not considered.

(3) Cascaded matching

The strategy of cascaded matching is used to improve matching accuracy, mainly because when a target is occluded for a long time, the uncertainty of Kalman filtering greatly increases, leading to a dispersion of continuous prediction probabilities. Assuming the original covariance matrix is normally distributed, continuous predictions without updates will increase the variance of this normal distribution, so points far from the mean in Euclidean distance may obtain the same Mahalanobis distance value as points closer in the previous distribution. In the final stage, the authors use *IOU* association from the previous SORT algorithm to match $n = 1$ unconfirmed and unmatched trajectories. This can alleviate significant changes caused by abrupt appearance shifts or partial occlusions. However, this approach may also connect some newly generated trajectories to older ones.

2.3.2 Deep appearance features

The original algorithm uses a residual convolutional neural network to extract the appearance features of the target, training the model on a large-scale pedestrian re-identification dataset for pedestrian detection and tracking. Since the original algorithm was only used for the pedestrian category and the input images were

scaled to 128×64 , which does not match the aspect ratio of vehicle targets, this article improves the network model by adjusting the input image size to 128×128 , as shown in Table 3. The adjusted network is then re-identification trained on the vehicle re-identification dataset VeRi [50].

3 Vehicle speed measurement

3.1 Model assumptions

All locations in road monitoring images can be mapped to the $Z_w = 0$ plane of the world coordinate system through camera calibration, as shown in Figure 5. However, the precise measurement of vehicle speed depends not only on camera calibration but also significantly on the vehicle's trajectory. To better implement vehicle speed measurement, the speed model assumes the following: (1) In highway scenarios, the road is relatively flat without significant undulations, meeting the condition of $Z_w = 0$; (2) In highway monitoring scenarios, the movement of vehicles between each frame is linear, allowing for the measurement of vehicles moving in both straight and non-straight paths using the proposed speed measurement method; (3) In highway video surveillance, the time interval between each frame is the same, facilitating the calculation of vehicle speed after obtaining the exact vehicle position using the interval between frames.

3.2 Model design and implementation

Based on the assumptions and establishment of the aforementioned speed model, the specific process of speed detection is implemented. Firstly, using the YOLO object detection algorithm, the coordinates of the top-left corner of the image detection box are obtained. By determining the length and width of the detection box, the coordinates of the center of the bottom edge of the box can be obtained. This ensures that the measured vehicle speed is closer to the actual speed. For every target vehicle in each frame of the video stream, a set of vector relations can be obtained, as shown in Eq. 10.

$$d_i = u_i(t) - u_i(t - \Delta t) \quad (10)$$

Here, $u_i(t)$ represents the center coordinates of the bottom edge of the vehicle target detection box in the current video frame; $u_i(t - \Delta t)$ represents the center coordinates of the bottom edge of the vehicle target detection box in the previous frame; Δt is the time interval between the two frames; $i = (1, 2, \dots, n)$ represents the tracked trajectory points.

d_i represents the pixel distance between adjacent frames, and calculating the speed requires mapping the pixel coordinates to world coordinates. The current common method involves camera calibration, but camera calibration requires knowledge of the camera's focal length, height, internal parameters, etc., and the calibration process can be cumbersome.

In This article, state estimation is performed using the popular methods of maximum likelihood estimation, maximum *a posteriori*

estimation, and non-linear least squares, selecting the best estimation parameters based on the loss in state estimation.

(1) Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) is an important and widely used method for estimating quantities. MLE explicitly uses a probability model with the goal of finding a system occurrence tree that can produce observed data with a high probability. MLE is a representative of a class of system occurrence tree reconstruction methods based entirely on statistics. Given a set of data, if we know it is randomly taken from a certain distribution, but we don't know the specific parameters of this distribution, that is, "the model is determined, but the parameters are unknown." For example, we know the distribution is a normal distribution, but we don't know the mean and variance; or it's a binomial distribution, but we don't know the mean. MLE can be used to estimate the parameters of the model. The objective of MLE is to find a set of parameters that maximize the probability of the model producing the observed data, as shown in Eq. 11.

$$\underset{\mu}{\operatorname{argmax}} p(X; \mu) \quad (11)$$

Here, $X = \{x_1, x_2, \dots, x_n\}$ represents the observed sequence data, and $p(X; \mu)$ is the likelihood function, which denotes the probability of the observed data occurring under the parameter μ . Assuming each observation is independent, as shown in Eq. 12.

$$p(x_1, x_2, \dots, x_n; \mu) = \prod_{i=1}^n p(x_i; \mu) \quad (12)$$

To facilitate differentiation, the log is generally taken of the target. Therefore, optimizing the likelihood function is equivalent to optimizing the log-likelihood function, as shown in Eqs 13, 14.

$$\underset{\mu}{\operatorname{argmax}} p(X; \mu) = \underset{\mu}{\operatorname{argmax}} \log p(X; \mu) \quad (13)$$

$$x_{MLE}^* = \operatorname{argmax} P(u | X) \quad (14)$$

(2) Maximum A Posteriori Estimation

In Bayesian statistics, Maximum A Posteriori (MAP) Estimation refers to the mode of the posterior probability distribution. MAP estimation is used to estimate the values of quantities that cannot be directly observed in experimental data. It is closely related to the classical method of Maximum Likelihood Estimation (MLE), but it uses an augmented optimization objective that further considers the prior probability distribution of the quantity being estimated. Therefore, MAP estimation can be seen as a regularized form of MLE, as shown in Eqs 15, 16.

$$\begin{aligned} \hat{\theta}_{MAP} &= \operatorname{argmax}_{\theta} p(\theta | x) \\ &= \operatorname{argmax}_{\theta} \frac{p(x | \theta) \times p(\theta)}{P(x)} \end{aligned} \quad (15)$$

$$= \operatorname{argmax}_{\theta} p(x | \theta) \times p(\theta)$$

$$x_{MAP}^* = \operatorname{argmax} P(x | z) = \operatorname{argmax} P(z | x)P(x) \quad (16)$$

Here, θ is the parameter to be estimated, and $p(\theta | x)$ represents the probability of occurrence of x when the estimated parameter is θ .

(3) Non-Linear Least Squares

The Least Squares Method (also known as the Method of Least Squares) is a mathematical optimization technique. It finds the best function match for data by minimizing the sum of the squares of the errors. The Least Squares Method can be used to easily obtain unknown data, ensuring that the sum of the squares of the errors between these obtained data and the actual data is minimized. The Least Squares Method can also be used for curve fitting, and other optimization problems can be expressed using this method by minimizing energy or maximizing entropy. Using the Least Squares Method to estimate the mapping relationship, the mapping parameters are obtained, as shown in Eqs 17, 18.

$$\min_x \sum \|y_i - f(x_i)\|_{\sum_i}^2 \quad (17)$$

Where $f(x_i)$ is a nonlinear function, and \sum_i^{-1} is the covariance matrix.

$$\psi(x) = \sum \|y_i - f(x_i)\|_{\sum_i}^2 \quad (18)$$

Then, the Gauss-Newton method is used to solve for $\psi(x)$, as shown in Eq. 19:

$$\begin{aligned} \psi(x) &= \sum \|y_i - f(x_i)\|_{\sum_i}^2 = \sum_{i=1}^m \|e_i(x)\|^2 = e_i^T(x) e_i(x) \\ &= \sum_{i=1}^m \varphi_i(x) \end{aligned} \quad (19)$$

For the sum of errors, we investigate the i term, also performing a second-order Taylor expansion, followed by differentiation. We first calculate its first-order derivative (gradient) and second-order derivative.

First-order derivative, as shown in Eqs 20, 21.

$$\frac{\partial \varphi_i(x)}{\partial x_j} = 2 \cdot e_i(x) \cdot \frac{\partial e_i(x)}{\partial x_j} \quad (20)$$

$$\frac{\partial \psi(x)}{\partial x_j} = \sum_{i=1}^m 2 \cdot e_i(x) \cdot \frac{\partial e_i(x)}{\partial x_j} \quad (21)$$

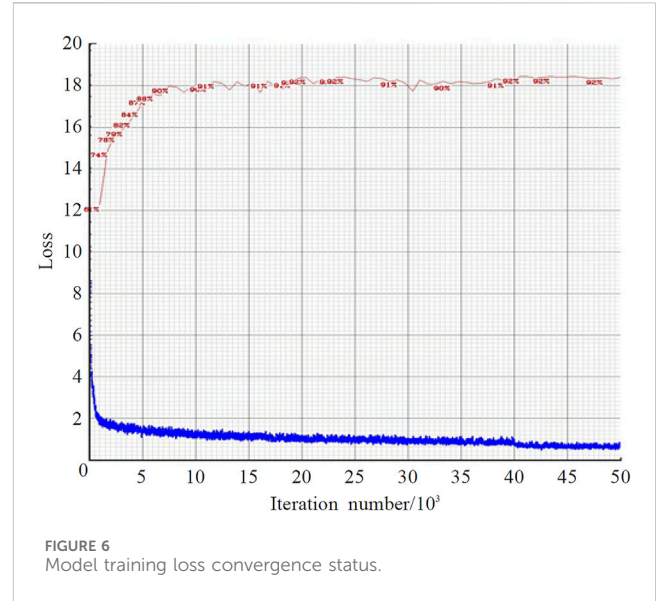
Where $\frac{\partial e_i(x)}{\partial x_j}$ is the element in the i column of the j row of the Jacobian matrix, thus the first-order derivative can also be expressed in the following form, as shown in Eq. 22.

$$\frac{\partial \psi(x)}{\partial x_j} = 2 \cdot J^T \cdot e(x) \quad (22)$$

Second-order derivative, as shown in Eq. 23.

$$\begin{aligned} \frac{\partial^2 \psi(x)}{\partial x_j \partial x_k} &= \frac{\partial}{\partial x_k} \left(\sum_{i=1}^m 2 \cdot e_i(x) \cdot \frac{\partial e_i(x)}{\partial x_j} \right) \\ &= 2 \sum_{i=1}^m \left(\frac{\partial e_i(x)}{\partial x_j} \cdot \frac{\partial e_i(x)}{\partial x_k} + e_i(x) \cdot \frac{\partial^2 e_i(x)}{\partial x_j \partial x_k} \right) \end{aligned} \quad (23)$$

Observing the result of the second-order derivative, the terms $\frac{\partial e_i(x)}{\partial x_j}$ and $\frac{\partial e_i(x)}{\partial x_k}$ are elements of the Jacobian matrix. When the iterative point is far from the target point, both the error and its second-order derivative are small and can be ignored. Therefore, the second-order derivative can be expressed in the following form, as shown in Eq. 24.



$$\frac{\partial^2 \psi(x)}{\partial x_j \partial x_k} = 2 \cdot J^T \cdot J \quad (24)$$

Therefore, after the second-order expansion, $\psi(x)$ can be written in the following form, as shown in Eq. 25:

$$\psi(x) = \psi(x^{(k)}) + 2(x - x^{(k)})^T J e(x) + (x - x^{(k)})^T J^T J (x - x^{(k)}) \quad (25)$$

Similarly, by differentiating it and setting the derivative equal to zero, Eq. 26:

$$\nabla \psi(x) = 2J^T e(x^{(k)}) + 2J^T J (x - x^{(k)}) = 0 \quad (26)$$

Let $\Delta x = x - x^{(k)}$ then, as shown in Eq. 27:

$$\Delta x = -(J^T J)^{-1} \cdot J^T \cdot e \quad (27)$$

3.3 Vehicle speed measurement

Through prior estimation, $u_i(t)$ and $u_i(t - \Delta t)$ can be mapped to the world coordinate system, representing the actual distance moved by the target vehicle from the previous frame to the current frame, as shown in Eq. 28. $\|S_i\|$ is measured in meters and is the Euclidean norm of S_i , representing the physical distance moved by the target vehicle in the world coordinate system from time $t - \Delta t$ to t . The speed of the vehicle target can be measured using $\|S_i\|$ as Eq. 29. Here, Δt is the time between two frames, measured in seconds, and is considered constant, being the reciprocal of the frame rate. For highway surveillance videos, which typically have a frame rate of 25 fps, $\Delta t = 1/25$.

$$S_i = \varphi(a, b, c) \cdot u_i(t) - \varphi(a, b, c) \cdot u_i(t - \Delta t) \quad (28)$$

$$v_i = \frac{\|S_i\|}{\Delta t} = \frac{\|\varphi(a, b, c) \cdot u_i(t) - \varphi(a, b, c) \cdot u_i(t - \Delta t)\|}{\Delta t} \quad (29)$$

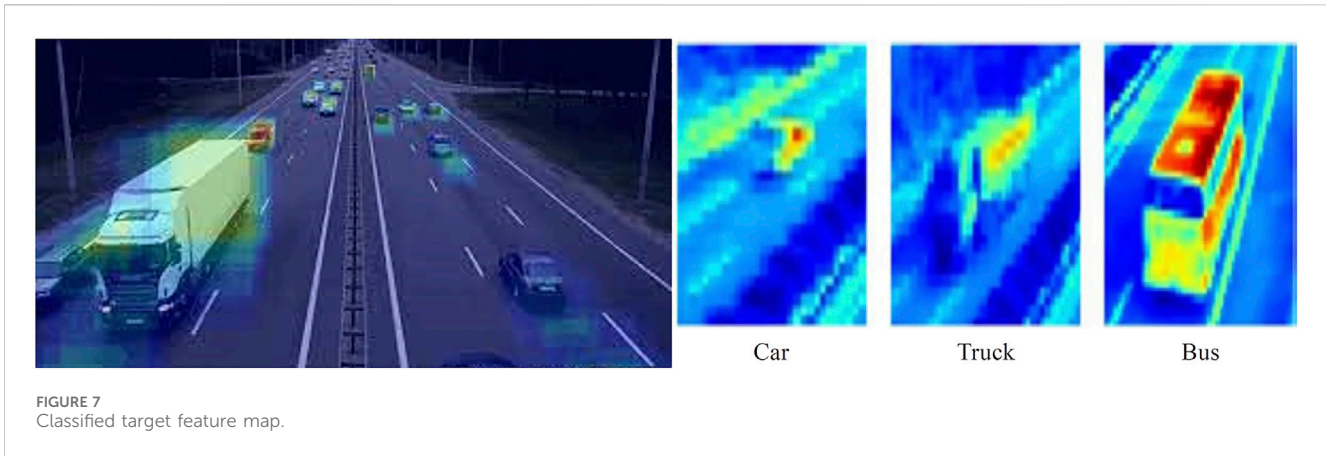


FIGURE 7
Classified target feature map.

Assuming a vehicle's trajectory contains m frame trajectory points, meaning in the first m frames of the video, the vehicle's speed between each adjacent pair of frames is v_1, v_2, \dots, v_{m-1} , then according to Eq. 29, v_1, v_2, v_{m-1} as shown in Eqs 30–32:

$$v_1 = \frac{\|S_1\|}{\Delta t} = \frac{\|\varphi(a, b, c) \cdot u_2(t) - \varphi(a, b, c) \cdot u_1(t)\|}{\Delta t} \quad (30)$$

$$v_2 = \frac{\|S_2\|}{\Delta t} = \frac{\|\varphi(a, b, c) \cdot u_3(t) - \varphi(a, b, c) \cdot u_2(t)\|}{\Delta t} \quad (31)$$

$$v_{m-1} = \frac{\|S_{m-1}\|}{\Delta t} = \frac{\|\varphi(a, b, c) \cdot u_m(t) - \varphi(a, b, c) \cdot u_{m-1}(t)\|}{\Delta t} \quad (32)$$

Therefore, the average driving speed of the target vehicle in the first m frames is as shown in Eq. 33. The detection of the target vehicle's speed is achieved by calculating the average of the instantaneous speeds over multiple frames.

$$v = \frac{\sum_{i=1}^{m-1} v_i}{m-1} \quad (33)$$

4 Model training and evaluation metrics selection

4.1 Experimental environment and model training

Experimental setup and hardware environment for the dataset: System Type: Windows 10 64-bit Operating System, Memory: 64GB, GPU: NVIDIA GeForce RTX3080ti, 24 GB Graphics Card. Software environment: The auxiliary environment includes CUDA V11.2, OpenCV4.5.3. This article tested different corresponding datasets for various traffic scenarios. The dataset established in This article comprises a total of 30,000 images, including a diverse collection from different scenes, angles, and times.

During training, 80% of the dataset was used for training, while 20% of the data was reserved for testing. Data augmentation was applied in this study, which involved random scaling, cropping, and arrangement of images using the Mosaic method. Random rotation (parameter set to 0.5), random exposure (parameter set to 1.5), and saturation (parameter set to 1.5) were employed to enrich the training data. The learning rate was initially set to 0.001, and the maximum number of training iterations was set to 50,000. To optimize model convergence, the

learning rate was adjusted to 0.0005 after 40,000 iterations. The input images to the network were resized to a resolution of 416×416 , and a batch size of 8 was used during training to ensure efficient network processing. The convergence of the model's training loss and mAP (mean Average Precision) can be observed in Figure 6. It shows that the model converged around 3,000 iterations, and as the loss decreased, mAP also reached a high level.

Convolutional Neural Networks (CNNs) are capable of extracting key features from image objects. The detected objects are classified into three categories: Car, Truck, and Bus. The unique features of each class can be observed in Figure 7, where each class of object exhibits distinct characteristics within the convolutional network. These distinct features are used for classification and detection purposes.

4.2 Selection of evaluation metrics

To verify the effectiveness of the model's detection, several typical metrics in the field of object detection and classification were selected for evaluation. For distracted driving behavior detection and classification, the focus is on detection precision and recall rate, as well as classification accuracy. Therefore, the model is evaluated using precision, recall, and F1_Score.

AP (Average Precision) is the average accuracy and a mainstream evaluation metric for object detection models. To correctly understand AP, it is necessary to use three concepts: Precision, Recall, and *IoU* (Intersection over Union). *IoU* measures the degree of overlap between two areas, specifically the overlap rate between the target window generated by the model and the originally marked window, which represents the detection accuracy *IoU*. The calculation formula is shown in Eq. 34. In an ideal situation, *IoU* equals 1, indicating a perfect overlap.

$$IoU = \frac{\text{Detection Result} \cap \text{Ground Truth}}{\text{Detection Result} \cup \text{Ground Truth}} \quad (34)$$

Precision and Recall in object detection: Assuming a set of images containing several targets for detection, Precision represents the proportion of targets detected by the model that are actual target objects, while Recall represents the proportion of all real targets detected by the model. TP (True Positive) denotes samples correctly identified as positive, TN (True Negative) denotes samples correctly identified as negative, FP (False Positive) denotes samples incorrectly identified as

TABLE 4 Fitting model results.

Number	Formulas	Abbreviation	AIC	BIC	R^2	p -value
1	$y = a^*x + b$	Line2p	139.69	141.6	0.774	3.3085e-05
2	$y = 1/(a^*x + b)$	Com2p	95.8	98.3	0.912	1.08e-08
3	$y = 1/(a^*x + b) + c$	Com3p	94.5	96.7	0.913	5.35e-07
4	$y = a^*x^2 + b^*x + c$	Line3p	118.0	121.0	0.958	2.59e-08
5	$y = a^* \ln(x) + b$	Log2p	130.9	132.8	0.880	7.2456e-07
6	$y = a^* \exp(b^*x)$	Exp2p	84.7	86.6	0.996	1.71e-15
7	$y = a^* \exp(b^*x) + c$	Exp3p	82.8	85.4	0.997	2.52e-14

Bold values represent the method chosen in this article.

positive, and FN (False Negative) denotes samples incorrectly identified as negative. The calculation of Precision and Recall values relies on the formulas shown in Eqs 35, 36.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (35)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (36)$$

After calculating values using the formula, a PR (Precision-Recall) curve can be plotted. The AP (Average Precision) is the mean of Precision values on the PR curve. To achieve more accurate results, the PR curve is smoothed, and the area under the smoothed curve is calculated using integral methods to determine the final AP value. The calculation formula is shown as Eq. 37.

$$AP = \int_0^1 P_{smooth}(r) dr \quad (37)$$

The F1-Score, also known as the F1 measure, is a metric for classification problems, often used as the final metric in multi-class problems. It is the harmonic mean of precision and recall. For the F1-Score of a single category, the calculation formula is as shown in Eq. 38.

$$F1_k = 2 \frac{\text{Recall}_k \times \text{Precision}_k}{\text{Recall}_k + \text{Precision}_k} \quad (38)$$

Subsequently, calculate the average value for all categories, denoted as F1. The calculation formula is shown in Eq. 39.

$$F1 = \left(\frac{1}{n} \sum F1_k \right)^2 \quad (39)$$

mAP (mean Average Precision) involves calculating the AP (Average Precision) for all categories and then computing the mean. The calculation formula is shown in Eq. 40.

$$mAP = \frac{\sum AP_i}{n}, i = 1, 2, \dots, n \quad (40)$$

5 Results and discussion

5.1 Evaluation of object detection model results

Based on the aforementioned evaluation metrics, the trained object detection models are tested and assessed using the test sets

from the datasets. The algorithm shows good statistical accuracy for different vehicle types, with APs of Car, Bus, Truck being 93.58, 91.26, 90.05 respectively, mAP at 92.42, and F1_Score at 97. This is primarily due to the high visibility in tunnel and roadbed sections, where target features are more distinct, resulting in a more accurate model. Overall, the model's detection accuracy for buses is lower than for other categories, mainly because the sample size for buses is significantly smaller than for other categories. However, with a mean Average Precision (mAP) exceeding 90%, it demonstrates that the proposed model is reliable and fully applicable to highway scenarios.

5.2 Evaluation of speed estimation results

5.2.1 Selection of optimal fitting model

Based on the data distribution, This article selects 7 video points for fitting analysis with 7 sets of linear and nonlinear data. This curve relationship is not intuitively obvious but requires statistical testing. The optimal fitting model is chosen by comparing the degree of fit and its significance. The Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are two commonly used indicators for assessing model fitness, with smaller values indicating a better-fitting model. Therefore, before selecting a model, it is necessary to assess the AIC and BIC values for each model, including dependent and independent variables. Additionally, the goodness of fit R^2 and p -value are also key parameters for evaluating the quality of the fit. As the data distribution within the range of road video surveillance is essentially similar in terms of distance calibration, a random surveillance point is selected for the fitting analysis of the 7 formulas, with results as shown in Table 4.

From Table 4, it is evident that apart from linear fitting, the goodness of fit R^2 for all other methods is greater than 0.8. Among them, the *Exp3p* fitting shows the best performance, hence *Exp3p* is chosen as the formula for distance-speed fitting.

To obtain the best fitting parameters for *Exp3p*, employing Maximum Likelihood Estimation, Maximum A Posteriori Estimation, and Non-linear Least Squares method for parameter estimation on the distance calibration data from 7 video points. The parameters are evaluated using AIC, BIC, R^2 , and p -value, with the evaluation results presented in Table 5; Figure 8.

From the above table, it is clear that for the *Exp3p* parameter estimation of the 7 video points, Maximum Likelihood Estimation

TABLE 5 Parameter estimation results.

Number	MLE				MAP				NLS			
	AIC	BIC	R^2	p -value	AIC	BIC	R^2	p -value	AIC	BIC	R^2	p -value
1	80	82	0.998	2.52e-14	81	83	0.998	1.15e-14	82	85	0.997	2.52e-14
2	83	86	0.994	1.76e-10	84	87	0.994	4.32e-10	86	89	0.993	5.38e-10
3	81	84	0.996	5.81e-13	84	86	0.995	7.65e-13	84	87	0.995	8.26e-13
4	94	100	0.923	5.63e-09	96	103	0.913	4.25e-08	98	105	0.902	5.63e-08
5	83	91	0.983	8.54e-13	85	92	0.980	2.85e-12	85	93	0.975	3.16e-12
6	84	87	0.992	2.52e-10	85	88	0.995	5.15e-10	87	90	0.990	6.87e-10
7	82	85	0.995	4.84e-12	83	87	0.994	6.62e-12	86	89	0.993	5.36e-12

Bold values represent the method chosen in this article.

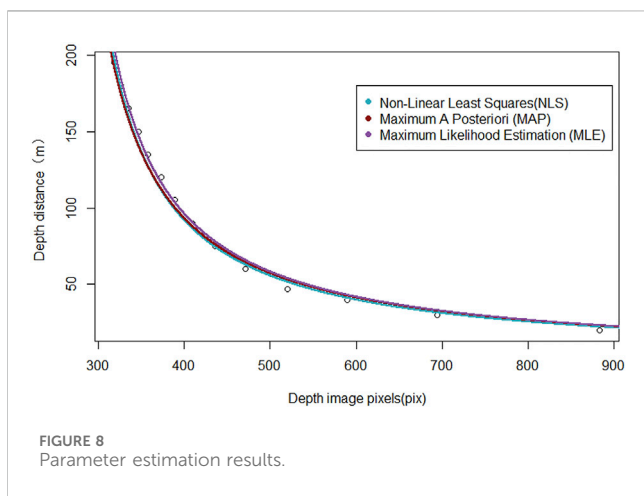


FIGURE 8 Parameter estimation results.

shows the best performance, followed by Maximum A Posteriori Estimation, and lastly Non-linear Least Squares method, as indicated by AIC, BIC, R^2 , and p -value.

5.2.2 Speed estimation results

To evaluate the measurement results of the speed estimation method, based on radar and video multi-sensor fusion technology, the results measured by millimeter-wave radar are taken as the true speed values. The verification experiment was conducted in the Shimen Tunnel on the Hanping Expressway in Shaanxi China,

where radar and video integration devices were installed at 150-m intervals, totaling seven units, to achieve holographic perception of traffic flow states within a 1050-m range, obtaining detailed information on coordinates, lane positions, and speeds for different lanes and vehicle types. Vehicle speeds detected by millimeter-wave radar and video were extracted using timestamps and target IDs. The comparison between the measured results and the true speed values, along with the overall experimental results and performance analysis, are shown in Table 6.

From Table 6, it is observed that the vehicle speed measurement method based on video, as discussed in This article, shows relatively good performance in scenarios with high overall speeds on highways. The minimum root mean square error is 2.0635, and the maximum is 9.2797. The main reasons for the larger deviation between the measured speeds and the actual values are environmental conditions, such as lighting and line shape. The coefficient of determination ranges from a minimum of 0.68259 to a maximum of 0.97730. The variation in the goodness of fit is for the same reasons as the minimum mean square error. Additionally, to further evaluate the speed tracking performance of this method, the vehicle speed measurement data from 7 video locations are manually divided into Front section, Middle section, Back section, and End section, for a comprehensive analysis of the overall tracking effect in these four segments, as seen in Figure 9.

As depicted in Figure 9, the effective measurement distance of this method is around 140 m, with the absolute speed error generally within 1–8 km/h, meeting the accuracy requirements for speed

TABLE 6 Overall speed measurement results and performance analysis.

Station number	MSE	RMSE	MAE	R^2
K733 + 953	26.9133	5.1878	3.8536	0.87993
K734 + 088	14.2012	3.7685	2.9179	0.90497
K734 + 843	52.2661	7.2295	5.3957	0.8889
K734 + 983	86.1127	9.2797	7.3639	0.68259
K735 + 123	6.6045	2.5699	2.0935	0.96168
K735 + 263	4.2581	2.0635	1.6984	0.97730
K735 + 403	81.6310	9.0205	7.6991	0.75010

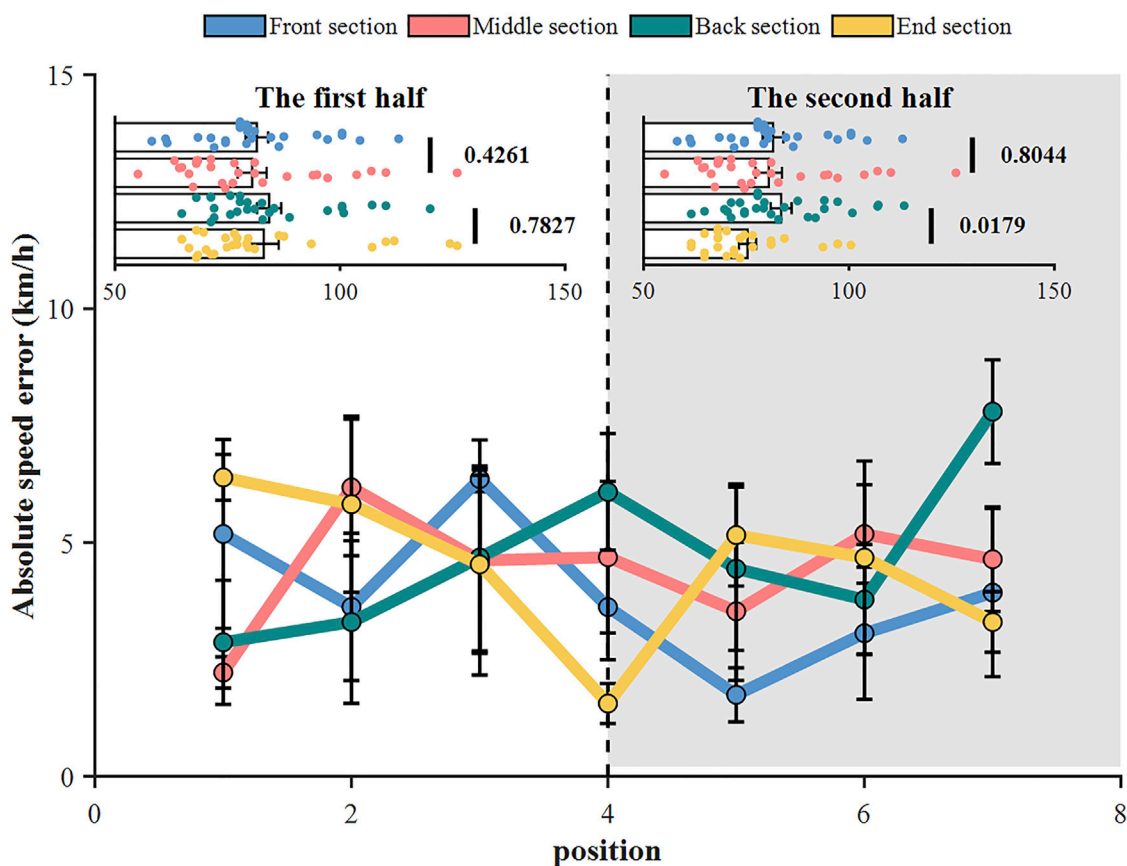


FIGURE 9
Analysis of speed tracking effect.

measurement. This method has certain advantages in distance detection, especially in tunnel scenarios, where a camera spacing of 150 m allows for continuous tracking of vehicle trajectories and speeds based on video. For further analysis of speed tracking differences within the 150 m detection range, it's divided into The first half and The second half. The first half data shows a minimum significance level of 0.4261, indicating small differences in speed tracking, reflecting stable tracking performance. The second half data has a minimum significance level of 0.0179, indicating some fluctuations in speed in the End section of The second half, but the absolute speed error still shows good precision.

6 Conclusion

This article proposes an improved YOLOv5s + DeepSORT vehicle speed measurement algorithm for surveillance videos in highway scenarios, capable of vehicle target detection and continuous speed tracking without camera prior parameters and calibration. The main conclusions are as follows:

- (1) The introduction of the Swin Transformer Block module improves the model's ability to capture local areas of interest, effectively increasing the detector's accuracy; using *CIoU* Loss to replace the original *GIoU* loss further enhances

the detector's localization precision and effectively reduces omissions in congested vehicle scenarios; the algorithm shows good statistical accuracy for different vehicle types, with APs of Car, Bus, Truck being 93.58, 91.26, 90.05 respectively, mAP at 92.42, and F1_Score at 97.

- (2) A calibration algorithm for traffic monitoring scenarios was proposed, which uses known reference points such as the image's centerline and contour marks. It applies Maximum Likelihood Estimation, Maximum A Posteriori Estimation, and Non-linear Least Squares method for the conversion between image pixel coordinates and actual coordinates. The parameter estimation showed good results, with Maximum Likelihood Estimation being the best, and AIC, BIC, R^2 , and p -value being 83.56, 87.86, and 8.66E-10 respectively.
- (3) The vehicle speed measurement is achieved by calculating the average of instantaneous speeds over multiple frames. This method's effective measurement distance is about 140m, with an absolute speed error generally within 1–8 km/h, meeting the accuracy requirements for speed measurement. It has certain advantages in distance detection, especially in tunnel scenarios where a camera spacing of 150 m allows for continuous tracking of vehicle trajectories and speeds based on video.
- (4) However, during experiments, it was found that vehicle speed accuracy is influenced by road geometry, environmental conditions, lighting, resolution, etc., These can be mitigated

through image enhancement optimization algorithms or by increasing video resolution, thus achieving more accurate vehicle speed measurements, which help regulatory bodies more effectively control speeds on the roads, reducing instances of speeding and thereby decreasing traffic accidents, enhancing road safety. Additionally, with the rapid development of multi-sensor fusion technology, the integration of video and millimeter-wave radar detection results can complement each other, providing technical support for active traffic safety management on highways.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

ZL: Conceptualization, Methodology, Writing—original draft. YB: Funding acquisition, Writing—original draft. XY: Project administration, Resources, Writing—review and editing. YL: Investigation, Writing—review and editing. SY: Software, Writing—review and editing. MW: Funding acquisition, Project administration, Writing—review and editing. QY: Data curation, Writing—review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research

References

- Wan Y, Huang Y, Buckles B. Camera calibration and vehicle tracking: highway traffic video analytics. *Transp Res C Emerg Technol* (2014) 44:202–13. doi:10.1016/j.trc.2014.02.018
- Karoń G, Mikulski J. Selected problems of transport modelling with ITS services impact on travel behavior of users. In: 2017 15th International Conference on ITS Telecommunications (ITST); 29–31 May 2017; Warsaw, Poland. IEEE (2017). p. 1–7. doi:10.1109/ITST.2017.7972231
- Wang Y, Yu C, Hou J, Chu S, Zhang Y, Zhu Y. ARIMA model and few-shot learning for vehicle speed time series analysis and prediction. *Comput Intell Neurosci* (2022) 2022:1–9. doi:10.1155/2022/2526821
- Jia S, Peng H, Liu S. Urban traffic state estimation considering resident travel characteristics and road network capacity. *J Transportation Syst Eng Inf Tech* (2011) 11: 81–5. doi:10.1016/S1570-6672(10)60142-0
- Javadi S, Dahl M, Pettersson MI. Vehicle speed measurement model for video-based systems. *Comput Electr Eng* (2019) 76:238–48. doi:10.1016/j.compeleceng.2019.04.001
- Dahl M, Javadi S. Analytical modeling for a video-based vehicle speed measurement framework. *Sensors (Switzerland)* (2020) 20(1):160. doi:10.3390/s20010160
- Khan A, Sarker DMSZ, Rayamajhi S. Speed estimation of vehicle in intelligent traffic surveillance system using video image processing. *Int J Sci Eng Res* (2014) 5(12): 1384–90. doi:10.14299/ijser.2014.12.003
- Wicaksono DW, Setiyono B. Speed estimation on moving vehicle based on digital image processing. *Int J Comput Sci Appl Math* (2017) 3(1):21–6. doi:10.12962/j24775401.v3i1.2117
- Lu S, Wang Y, Song H. A high accurate vehicle speed estimation method. *Soft Comput* (2020) 24:1283–91. doi:10.1007/s00500-019-03965-w
- Liu C, Huynh DQ, Sun Y, Reynolds M, Atkinson S. A vision-based pipeline for vehicle counting, speed estimation, and classification. *IEEE Trans Intell Transportation Syst* (2021) 22:7547–60. doi:10.1109/TITS.2020.3004066
- Bhardwaj R, Tummala GK, Ramalingam G, Ramjee R, Sinha P. AutoCalib: automatic traffic camera calibration at scale. *ACM Trans Sen Netw* (2018) 14(3-4):1–27. doi:10.1145/3199667
- Qimin X, Xu L, Mingming W, Bin L, Xianghui S. A methodology of vehicle speed estimation based on optical flow. In: Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics; 08–10 October 2014; Qingdao, China. IEEE (2014). p. 33–7. doi:10.1109/SOLI.2014.6960689
- Schoepflin TN, Dailey DJ. Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. *IEEE Trans Intell Transportation Syst* (2003) 4:90–8. doi:10.1109/TITS.2003.821213
- Han J, Heo O, Park M, Kee S, Sunwoo M. Vehicle distance estimation using a mono-camera for FCW/AEB systems. *Int J Automotive Tech* (2016) 17:483–91. doi:10.1007/s12239-016-0050-9
- Sochor J, Juranek R, Spanhel J, Marsik L, Siroky A, Herout A, et al. Comprehensive data set for automatic single camera visual speed measurement. *IEEE Trans Intell Transportation Syst* (2019) 20:1633–43. doi:10.1109/TITS.2018.2825609
- Lin H-Y, Li K-J, Chang C-H. Vehicle speed detection from a single motion blurred image. *Image Vis Comput* (2008) 26:1327–37. doi:10.1016/j.imavis.2007.04.004
- Celik T, Kusetogullari H. Solar-powered automated road surveillance system for speed violation detection. *IEEE Trans Ind Elect* (2010) 57:3216–27. doi:10.1109/TIE.2009.2038395
- Nguyen TT, Pham XD, Song JH, Jin S, Kim D, Jeon JW. Compensating background for noise due to camera vibration in uncalibrated-camera-based vehicle

Acknowledgments

The authors would like to thank the support of their colleagues in the Research and Development Center of Transport Industry of Self-driving Technology.

Conflict of interest

Authors ZL, XY, SY, MW, and QY were employed by China Merchants Chongqing Communications Research and Design Institute Co., Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- speed measurement system. *IEEE Trans Veh Technol* (2011) 60:30–43. doi:10.1109/TVT.2010.2096832
19. Eslami H, Raie AA, Faez K. Precise vehicle speed measurement based on a hierarchical homographic transform estimation for law enforcement applications. *IEICE Trans Inf Syst* (2016) E99.D:1635–44. doi:10.1587/transinf.2015EDP7371
20. Famouri M, Azimifar Z, Wong A. A novel motion plane-based approach to vehicle speed estimation. *IEEE Trans Intell Transportation Syst* (2019) 20:1237–46. doi:10.1109/TITS.2018.2847224
21. Li J, Chen S, Zhang F, Li E, Yang T, Lu Z. An adaptive framework for multi-vehicle ground speed estimation in airborne videos. *Remote Sens (Basel)* (2019) 11:1241. doi:10.3390/rs11101241
22. Koyuncu H, Koyuncu B. Vehicle Speed detection by using Camera and image processing software. *Int J Eng Sci (Ghaziabad)* (2018) 7:64–72. doi:10.9790/1813-0709036472
23. Kim J-H, Oh W-T, Choi J-H, Park J-C. Reliability verification of vehicle speed estimate method in forensic videos. *Forensic Sci Int* (2018) 287:195–206. doi:10.1016/j.forsciint.2018.04.002
24. Sochor J, Juránek R, Herout A. Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement. *Computer Vis Image Understanding* (2017) 161:87–98. doi:10.1016/j.cviu.2017.05.015
25. Palubinskas G, Kurz F, Reinartz P. Model based traffic congestion detection in optical remote sensing imagery. *Eur Transport Res Rev* (2010) 2:85–92. doi:10.1007/s12544-010-0028-z
26. Doğan S, Temiz MS, Külür S. Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera. *Sensors* (2010) 10(5):4805–24. doi:10.3390/s100504805
27. Li S, Yu H, Zhang J, Yang K, Bin R. Video-based traffic data collection system for multiple vehicle types. *IET Intell Transport Syst* (2014) 8:164–74. doi:10.1049/iet-its.2012.0099
28. Jeyabharathi D, Dejeu DD. Vehicle tracking and speed measurement system (VTSM) based on novel feature descriptor: diagonal hexadecimal pattern (DHP). *J Vis Commun Image Represent* (2016) 40:816–30. doi:10.1016/j.jvcir.2016.08.011
29. Agrawal SC, Tripathi RK. An image processing based method for vehicle speed estimation. *Int J Scientific Tech Res* (2020) 9:1241–6.
30. Biswas D, Su H, Wang C, Stevanovic A. Speed estimation of multiple moving objects from a moving UAV platform. *ISPRS Int J Geoinf* (2019) 8(6):259. doi:10.3390/ijgi8060259
31. Lee J, Roh S, Shin J, Sohn K. Image-based learning to measure the space mean speed on a stretch of road without the need to tag images with labels. *Sensors (Switzerland)* (2019) 19:1227. doi:10.3390/s19051227
32. Dong H, Wen M, Yang Z. Vehicle speed estimation based on 3D ConvNets and non-local blocks. *Future Internet* (2019) 11(6):123. doi:10.3390/fi11060123
33. Luvizon DC, Nassu BT, Minetto R. A video-based system for vehicle speed measurement in urban roadways. *IEEE Trans Intell Transportation Syst* (2017) 18:1–12. doi:10.1109/TITS.2016.2606369
34. Yang L, Li M, Song X, Xiong Z, Hou C, Qu B. Vehicle speed measurement based on binocular stereovision system. *IEEE Access* (2019) 7:106628–41. doi:10.1109/ACCESS.2019.2932120
35. Blankenship K, Diamantas S. Detection, tracking, and speed estimation of vehicles: a homography-based approach. *IMPROVE* (2022) 1:211–8. doi:10.5220/0011093600003209
36. Fernández Llorca D, Hernández Martínez A, García Daza I. Vision-based vehicle speed estimation: a survey. *IET Intell Transport Syst* (2021) 15:987–1005. doi:10.1049/itr2.12079
37. Kim HJ. Vehicle detection and speed estimation for automated traffic surveillance systems at nighttime. *Tehnicki Vjesnik* (2019) 26:091448. doi:10.17559/TV-20170827091448
38. Ashraf MH, Jabeen F, Alghamdi H, Zia MS, Almutairi M. HVD-net: a hybrid vehicle detection network for vision-based vehicle tracking and speed estimation. *J King Saud Univ - Comp Inf Sci* (2023) 35:101657. doi:10.1016/j.jksuci.2023.101657
39. Pal SK, Pramanik A, Maiti J, Mitra P. Deep learning in multi-object detection and tracking: state of the art. *Appl Intelligence* (2021) 51:6400–29. doi:10.1007/s10489-021-02293-7
40. Jiao L, Zhang F, Liu F, Yang S, Li L, Feng Z, et al. A survey of deep learning-based object detection. *IEEE Access* (2019) 7:128837–68. doi:10.1109/ACCESS.2019.2939201
41. Khosravi H, Dehkordi RA, Ahmadyfard A. Vehicle speed and dimensions estimation using on-road cameras by identifying popular vehicles. *Scientia Iranica* (2022) 29. doi:10.24200/sci.2020.55331.4174
42. Huang L, Zhe T, Wu J, Wu Q, Pei C, Chen D. Robust inter-vehicle distance estimation method based on monocular vision. *IEEE Access* (2019) 7:46059–70. doi:10.1109/ACCESS.2019.2907984
43. Jamshidnejad A, De Schutter B. Estimation of the generalised average traffic speed based on microscopic measurements. *Transportmetrica A: Transport Sci* (2015) 11: 525–46. doi:10.1080/23249935.2015.1026957
44. Sarkar NC, Bhaskar A, Zheng Z, Miska MP. Microscopic modelling of area-based heterogeneous traffic flow: area selection and vehicle movement. *Transp Res Part C Emerg Technol* (2020) 111:373–96. doi:10.1016/j.trc.2019.12.013
45. Appathurai A, Sundarasekar R, Raja C, Alex EJ, Palagan CA, Nithya A. An efficient optimal neural network-based moving vehicle detection in traffic video surveillance system. *Circuits Syst Signal Process* (2020) 39:734–56. doi:10.1007/s00034-019-01224-9
46. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, et al. Swin transformer: hierarchical vision transformer using shifted windows. *Proc IEEE Int Conf Comp Vis* (2021) 10012–22. doi:10.1109/ICCV48922.2021.00986
47. Zhang Q, Zhang M, Chen T, Sun Z, Ma Y, Yu B. Recent advances in convolutional neural network acceleration. *Neurocomputing* (2019) 323:37–51. doi:10.1016/j.neucom.2018.09.038
48. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and realtime tracking. In: Proceedings - International Conference on Image Processing, ICIP; 17–20 September 2017; Beijing, China. IEEE (2016). p. 3464–8. doi:10.1109/ICIP.2016.7533003
49. Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. In: Proceedings - International Conference on Image Processing, ICIP; 17–20 September 2017; Beijing, China. IEEE (2017). p. 3645–9. doi:10.1109/ICIP.2017.8296962
50. Liu X, Liu W, Mei T, Ma H. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In: *Lecture notes in computer science including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics*. Berlin, Germany: Springer (2016). p. 869–84. doi:10.1007/978-3-319-46475-6_53