frontiers in Physics

ORIGINAL RESEARCH
published: 21 February 2022
doi: 10.3389/fphy.2022.829734

A Deep Learning Framework for Video-Based Vehicle Counting

Haojia Lin[1,2], *Zhilu Yuan*[2], *Biao He*[2,3,4], *Xi Kuai*[2], *Xiaoming Li*[2] *and Renzhong Guo*[1,2]*

[1]School of Resource and Environmental Sciences, Wuhan University, Wuhan, China, [2]Guangdong–Hong Kong-Macau Joint Laboratory for Smart Cities, and Shenzhen Key Laboratory of Digital Twin Technologies for Cities, and Research Institute for Smart Cities, School of Architecture and Urban Planning, Shenzhen University, Shenzhen, China, [3]MNR Technology Innovation Center of Territorial & Spatial Big Data, Shenzhen, China, [4]MNR Key Laboratory of Urban Land Resources Monitoring and Simulation, Shenzhen, China

Traffic surveillance can be used to monitor and collect the traffic condition data of road networks, which plays an important role in a wide range of applications in intelligent transportation systems (ITSs). Accurately and rapidly detecting and counting vehicles in traffic videos is one of the main problems of traffic surveillance. Traditional video-based vehicle detection methods, such as background subtraction, frame difference, and optical flow have some limitations in accuracy or efficiency. In this paper, deep learning is applied for vehicle counting in traffic videos. First, to solve the problem of the lack of annotated data, a method for vehicle detection based on transfer learning is proposed. Then, based on vehicle detection, a vehicle counting method based on fusing the virtual detection area and vehicle tracking is proposed. Finally, due to possible situations of missing detection and false detection, a missing alarm suppression module and a false alarm suppression module are designed to improve the accuracy of vehicle counting. The results show that the proposed deep learning vehicle counting framework can achieve lane-level vehicle counting without enough annotated data, and the accuracy of vehicle counting can reach up to 99%. In terms of computational efficiency, this method has high real-time performance and can be used for real-time vehicle counting.

Keywords: intelligent transportation systems, traffic video, vehicle detection, vehicle counting, deep learning

OPEN ACCESS

Edited by:
Chao Gao,
Southwest University, China

Reviewed by:
Yu Lin,
Ningbo University, China
Junjie Pang,
Qingdao University, China
Qiuyang Huang,
Jilin University, China

***Correspondence:**
Renzhong Guo
guorz@szu.edu.cn

Specialty section:
This article was submitted to Social Physics, a section of the journal Frontiers in Physics

Received: *06 December 2021*
Accepted: *11 January 2022*
Published: *21 February 2022*

Citation:
Lin H, Yuan Z, He B, Kuai X, Li X and Guo R (2022) A Deep Learning Framework for Video-Based Vehicle Counting. Front. Phys. 10:829734. doi: 10.3389/fphy.2022.829734

1 INTRODUCTION

The rapid growth of the urban population and motor vehicles has led to a series of traffic problems. Intelligent transportation systems (ITSs) are considered the best tool to solve these problems. With the development of the Internet of Things (iot) technology, communications technology and computer vision, traffic surveillance has become a major technology of traffic parameter collection and plays a crucial role [1–3]. Traffic flow is an important basic parameter in ITS [4, 5], and accurately and rapidly detecting and counting vehicles based on traffic videos is a common research topic. Over the last decades, various vision approaches have been proposed to automatically count vehicles in traffic videos. Many existing vehicle counting methods rely on a vehicle detector based on the vehicle's appearance and features that are located *via* foreground detection, and vehicles are counted based on vehicle detection results [6–8]. In general, the methods for vehicle counting based on traffic videos can be divided into two subtasks: vehicle detection and vehicle counting.

In vehicle detection, the common methods include background subtraction [9, 10], frame difference [11, 12], optical flow [13, 14] and deep learning object detection [15–18]. The first three methods detect vehicles through manually extracted features, which are relatively simple, but

they also have some limitations in accuracy or robustness. Instead of manually extracting features, the deep learning method simulates information processing of the human brain and enables the constructed network to perform automatic feature extraction by training on a large annotated dataset [19]. However, this method relies on a large training dataset and is difficult to apply to various traffic video scenarios. Transfer learning can be combined with deep learning to build a model of target tasks based on source tasks [17, 20–22], but combining transfer learning with deep learning in the absence of annotated data is still an important research direction to study.

For vehicle counting, the usual approaches can be mainly divided into two categories: vehicle counting based on the virtual detection area [23–25] and vehicle counting based on vehicle tracking [26–28]. The virtual detection area sets up virtual detection areas in a video to determine whether there are vehicles passing through according to the change of the area's grey value, which is highly efficient, but it easily makes mistakes due to lane changing or parallel driving. Vehicle tracking extracts the trajectory of each vehicle by matching vehicles detected in each frame of video sequences and counts the number of vehicles based on vehicle trajectories, which has a high accuracy, but the computing cost is relatively high.

With the development of deep learning object detection and the computability of hardware, such as GPUs, it is possible to build a deep learning vehicle counting model with high accuracy and efficiency, but there are still some challenges. On the one hand, it is time-consuming to build a training dataset for a specific traffic scenario. The primary problem is constructing a vehicle detection model with good performance rapidly through transfer learning in the absence of training data. On the other hand, although deep learning vehicle detection can detect vehicles more accurately, it is still inevitable that situations of missing detection or false detection occur. Avoiding errors caused by these situations is a key problem to solve to further improve the accuracy of vehicle counting when the accuracy of vehicle detection is difficult to improve.

In this paper, a deep learning framework for vehicle counting is proposed. To solve the problem of lacking training data, a method of vehicle detection model construction based on transfer learning and open datasets is proposed, which can build a vehicle detection model with high-quality performance rapidly in the absence of training data. Moreover, for the possible situations of missing detection and false detection, a vehicle counting method based on fusing the virtual detection area and vehicle tracking is proposed, which can avoid the errors caused by missing detection or false detection and further improve the accuracy of vehicle counting.

2 PROPOSED FRAMEWORK

In this section, we introduce the proposed deep learning framework for video-based vehicle counting. As shown in **Figure 1**, the framework can be divided into three stages: dataset building, vehicle detection model construction, and vehicle counting. First, to build a training dataset for deep learning vehicle detection model construction and avoid spending too much time labelling images, annotated images containing vehicles in the open dataset are extracted as training data. In addition, to further improve and evaluate the performance of the vehicle detection model, a few frame images in traffic videos are extracted and divided into supplemental training data and testing data after being labelled. Next, in the vehicle detection model construction stage, a deep learning object detection model that meets the requirements in terms of accuracy and efficiency is utilized as the basic model, and instance-based transfer learning and parameter-based transfer learning are adopted to construct a vehicle detection model. Finally, in the vehicle counting stage, vehicles in each frame are detected by the deep learning vehicle detection model and counted by the vehicle counting model based on the fusion of virtual detection area and vehicle tracking, where a missing alarm suppression module based on vehicle tracking and a false alarm suppression module based on bounding box size statistics are designed to avoid the vehicle counting error caused by missing detection or false detection.

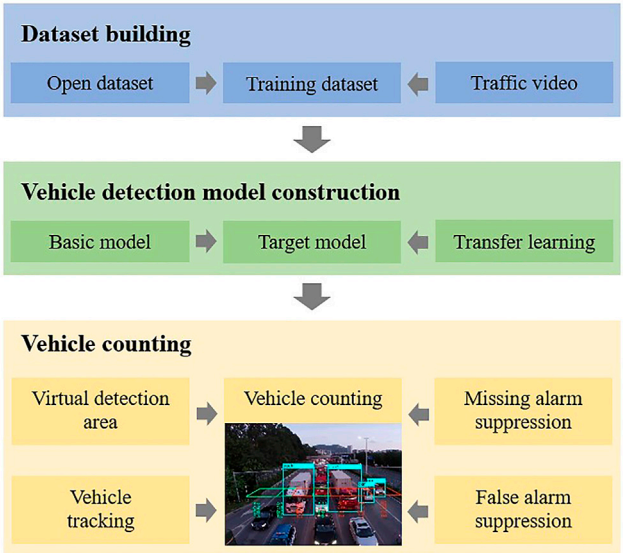

FIGURE 1 | Framework for video-based vehicle counting.

3 VEHICLE DETECTION

In the vehicle detection stage, a deep learning object detection method is adopted. Specifically, to solve the problem of lacking training data, transfer learning is used to construct a deep learning vehicle detection model. Employing a deep learning object detection model as the basic model and vehicle data in open datasets as the training data, the vehicle detection model is constructed by transfer learning.

3.1 Deep Learning Object Detection

In deep learning object detection models, a convolutional neural network (CNN) is used to extract image features, and then a

classifier and regressor are used to classify and locate the extracted features. The existing models can be mainly divided into two categories: two-stage detectors and one-stage detectors. Two-stage detectors consist of a region proposal network to feed region proposals into a classifier and regressor, which have high object recognition and localisation accuracy, but the efficiency is low, such as in the R-CNN [29], Fast R-CNN [30] and Faster R-CNN [31]. One-stage detectors regard object detection as a regression problem, which takes the entire image into a classifier and regressor and predicts objects directly without a region proposal step. Thus, they run faster than two-stage detectors, such as SSD [32] and YOLO [33–35].

In vehicle counting, accuracy and efficiency are both important. Therefore, a one-stage detector seems to be a better choice for vehicle detection model construction. As a typical representative of one-stage detectors, YOLO has better performance in terms of efficiency and accuracy than many other detectors, and the trade-off between accuracy and efficiency can be made according to the requirements [35]. Thus, in this framework, YOLO is selected as the basic model to construct a vehicle detection model. The implementation of YOLO is as follows:

1) A series of convolutional layers and residual layers are used to extract the features of the input image, and finally, three feature maps with different scales are obtained.
2) Each feature map is divided into S × S grids, and B anchor boxes are set for each grid cell to detect objects.
3) If the centre of an object falls into a grid cell, the classification probabilities (including the probability of each classification) and bounding box information (central point coordinate, width, height, and detected confidence) of that object would be detected by that grid cell.
4) The classification with maximum probability and the bounding box with maximum is taken to detect confidence as the output result of each grid cell, and the product of the corresponding classification probability and the bounding box is taken to detect confidence as the final detected confidence.
5) For each feature map, the above processes are carried out, and the final result is obtained by synthesising the results of three scales.

For object detection based on YOLO, the frame images in traffic video are taken as the input, and the detection result is taken as the input of vehicle counting. $BB_j = \{bb_i, i = 1, 2, \ldots, n\}$ represents the detection result of a frame, bb_i represents one bounding box of the detected object in the detection result and consists of six attributes: detected frame number f, central point coordinate (x, y), width w, height h, classification c, and detected confidence p.

3.2 Transfer Learning

Transfer learning aims to extract the knowledge from one or more source tasks and applies them to a target task [36]. According to the representation of transferred knowledge, transfer learning can be classified into four categories:

1) Instance-based transfer learning: Certain parts of data in the source task can be reused and combined with data in the target task to train a target model.
2) Parameter-based transfer learning: Some parameters, prior distributions, or hyperparameters of the models are shared between the source and the target task to improve the effectiveness of learning.
3) Feature-representation transfer learning: A good feature representation that reduces the difference between the source and target tasks is found and the knowledge used to transfer into the learned feature representation is encoded.
4) Relational-knowledge transfer learning: A mapping of relational knowledge between the source and target task is built, and the knowledge to be transferred is transformed into the relationship among the data.

3.3 Transfer Learning Based on YOLO

Traffic videos in this study are captured from fixed cameras above a straight road, including different light conditions, shooting directions, traffic conditions, and resolutions, where cars, buses, and trucks appear and the number of these three vehicle types are counted separately. In deep learning vehicle detection model construction, vehicle annotated data is needed as training data. Some open datasets, such as MS COCO [37], ImageNet [38] and PASCAL VOC [39], contain vehicle annotated data. Although they are different from vehicles in traffic videos, they can be used to build a target model through transfer learning. However, open datasets also have some limitations in this task because most of them are taken in the horizontal direction and fewer are taken from the top tilt down direction. Moreover, there are inevitably some annotation errors in some images. In addition, because of the difference between open datasets and traffic videos, the constructed model with high accuracy in open datasets may not perform well in traffic videos. Thus, it is necessary to build a supplemental dataset by labelling some traffic scenario images, and then using it as training data and testing data to further improve and evaluate the performance of the constructed vehicle detection model.

Based on YOLO and vehicle annotated data in open datasets, transfer learning is used to construct a deep learning vehicle detection model. The process is as follows:

1) Source model construction: YOLO trained by vehicle data in the MS COCO dataset is used as a source model. First, some annotated images containing vehicles (car, bus, and truck) in the MS COCO dataset are extracted. Then, k-means clustering is used on the bounding boxes of the extracted dataset, and nine anchor boxes (3 for each feature map in YOLO) for the model are obtained. Finally, the extracted dataset is used to train YOLO, and a source model Model-1 is obtained.
2) Supplemental dataset building: Model-1 is used to detect frame images in traffic videos, and the results are further processed into a refined annotated dataset. First, some frame images in traffic videos are extracted every fixed interframe space as a dataset to be annotated. Then, Model-1 is used to detect the extracted frame images and utilise the detection results as annotated data. Finally, the annotated data is further

processed to make the annotation more reliable, the false bounding boxes are removed, the missing detected objects are labelled, and the bounding boxes with inaccurate localisation and size are adjusted.

3) Transfer learning: Parameter-based transfer learning and instance-based transfer learning are adopted to construct a deep learning vehicle detection model. First, based on parameter-based transfer learning, the task of Model-1 and the target task are both considered vehicle detection, which have a strong correlation so that the parameters of Model-1 can be used to initialise the network of the target model. Then, based on instance-based transfer learning, the vehicle data in the MS COCO dataset are combined with the supplemental dataset, and k-means clustering is used on the combination to obtain another nine anchor boxes for YOLO. Finally, the merged dataset is used to train the initialised model, and a target model, Model-2, is obtained.
4) Target model optimization: Using training data in the supplemental dataset as training data, we further fine-tune the parameters of Model-2 to obtain a better target model Model-3.

4 VEHICLE COUNTING

In the vehicle counting stage, on the basis of vehicle bounding boxes obtained from vehicle detection, a new method of vehicle counting based on fusing virtual detection area and vehicle tracking is proposed, which considers the possible situation of missing detection and false detection, while combining the ideas of the traditional vehicle counting method based on virtual detection area and vehicle tracking.

4.1 Virtual Detection Area and Vehicle Counting

Considering the real-time requirement of vehicle counting, vehicle counting based on virtual detection area is employed as the basic method, and corresponding improvements are made for the input as a bounding box. The principle of the improved method is shown in **Figure 2**, and the process is as follows:

1) Virtual detection areas are set for each lane of the road section in the video. To ensure that at most one vehicle passes through the detection area at a time, the size of the detection area needs to have some restrictions. The length is close to the length of a car, and the width is similar to the width of the lane. In this case, there is not more than one vehicle passing through the detection area at the same time, which can reduce the complexity of vehicle tracking and vehicle counting.
2) The relative location relationship between the detected vehicle bounding box and the virtual detection area is calculated frame-by-frame, and the status of the detection area is updated. As shown in **Figure 2A**, if there is no central point of the bounding box located in the detection area, the status of that area is $S = 0$, indicating that no vehicle passes through. If there is a central point of the vehicle bounding box detected in the detection area, the status of that area is updated to $S = 1$, indicating that there is one vehicle passing through.
3) According to the status changes of each detection area in the frame sequence, the flow curve of each area can be obtained, and then the vehicle number can be counted. As shown in **Figure 2B**, when the status of an area in a frame is $S = 0$ and becomes $S = 1$ in the next frame, the vehicle number of that area is added to 1.

4.2 Missing Detection and False Detection

Missing detection (**Figure 3**, the yellow bounding box is missing detection) and false detection (**Figure 4**, the red bounding box is false detection) may occur in vehicle detection in continuous frame sequences, which cause vehicle counting errors. As shown in **Figure 5**, two vehicles pass through a virtual detection area, however, the first vehicle is determined to be a new vehicle when it is detected again after missing detection in one or several frames. Thus, the vehicle counting error occurs, and two vehicles are counted as three vehicles.

4.3 The Missing Alarm Suppression Module Based on Vehicle Tracking

To detect missing vehicles, considering that vehicle tracking has the ability to lock on each vehicle, a missing alarm suppression module based on vehicle tracking is added to the vehicle counting model. The module tracks detected vehicles in each area frame-by-frame and determines whether the detected passing vehicle in an area is a new vehicle to avoid incorrect counting caused by missing detection. The process is as follows:

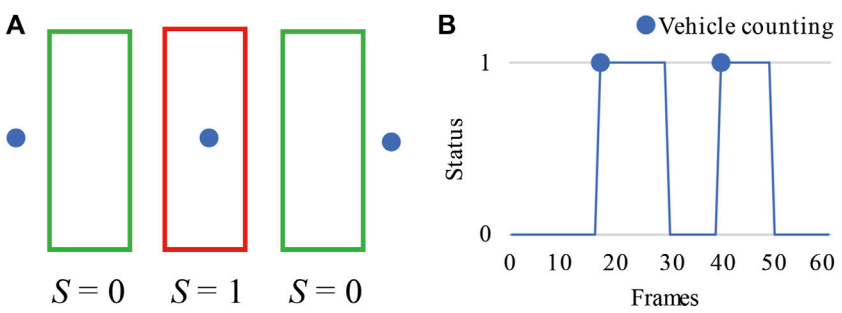

FIGURE 2 | The principle of vehicle counting based on virtual detection areas. **(A)** Status change of a virtual detection area; **(B)** Flow curve of a virtual detection area.

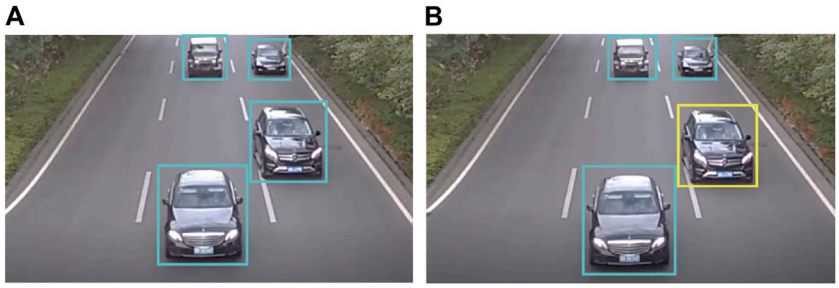

FIGURE 3 | Vehicle missing detection. **(A)** Results of vehicle detection in frame *i*; **(B)** Results of vehicle detection in frame *i* + 1.

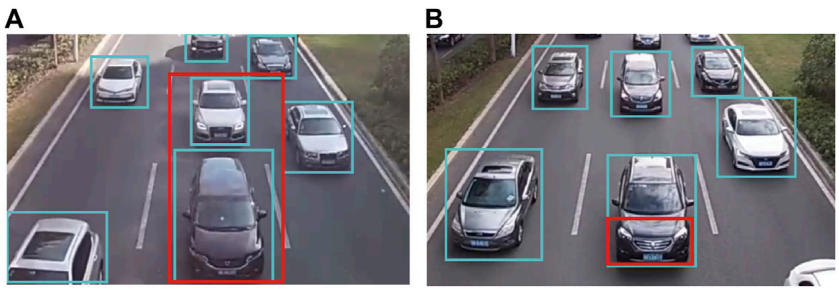

FIGURE 4 | Vehicle false detection. **(A)** Vehicle false detection situation a; **(B)** Vehicle false detection situation b.

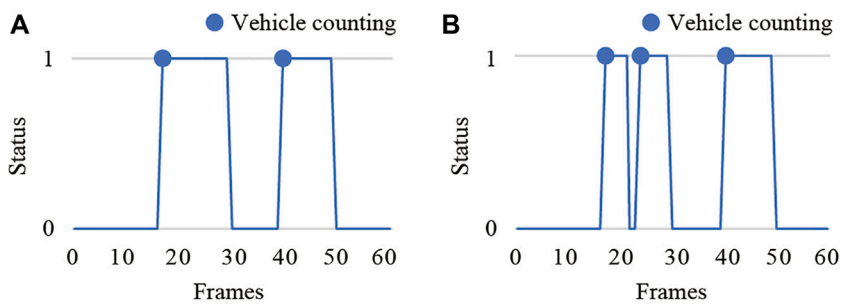

FIGURE 5 | Incorrect vehicle counting caused by vehicle missing detection. **(A)** Flow curve of correct counting; **(B)** Flow curve of incorrect counting.

1) Vehicles are detected in each frame. If the central point of a detected vehicle bounding box is located in an area, the bounding box of that vehicle is marked as the tracking vehicle of that area. bb_i^k is used to represent the detection result of area k in a frame, and tb_k is used to represent the tracking vehicle of area k.
2) Vehicles in the next frame are detected and match each detected vehicle with the tracking vehicle of the corresponding area according to the space-time distance of the vehicles. If there is a match, the tracking vehicle is updated to the detected vehicle. If no matching occurs, the detected vehicle is marked as a new tracking vehicle of the corresponding area.
3) Step 2 is carried out frame-by-frame, the vehicle number of each detection area is calculated based on the flow curve. In addition, to calculate the number of each vehicle type, such as car, bus, and truck, the flow curve of each vehicle type is generated according to the classification of the detected vehicles, and then each kind of number is calculated based on the corresponding kind of flow curve separately.

In vehicle tracking, to ensure the real-time performance of the module, an efficient vehicle tracking method based on the space-time distance of vehicles is used to match the detected vehicle with the tracking vehicle, including the space distance (*SD*) between central points of bounding boxes and the time distance (*TD*) of the bounding boxes detected frame number. The smaller the *SD* or *TD* is, the more likely two detected vehicles are the same vehicle. The calculation is as follows:

$$SD\left(bb_i^k,\ tb_k\right) = \frac{\sqrt{\left(x_{bb_i^k} - x_{tb_k}\right)^2 + \left(y_{bb_i^k} - y_{tb_k}\right)^2}}{\bar{h}_k^{car}} \quad (1)$$

$$TD\left(bb_i^k, tb_k\right) = f_{bb_i^k} - f_{tb_k} \quad (2)$$

where bb_i^k represents the bounding box detected in the current frame in area k, tb_k represents the bounding box of the tracking vehicle of area k, and $\bar{h}_k^{car}$ represents the average height of bounding boxes whose classification is the car in area k. Then, the new vehicle value (NV) is used to determine whether bb_i^k is a new vehicle or not. The calculation is as follows:

$$NV = \begin{cases} 1, & SD\left(bb_i^k,\ tb_k\right) > T_{SD}\ and\ TD\left(bb_i^k,\ tb_k\right) > T_{TD} \\ 0, & otherwise \end{cases} \quad (3)$$

where T_{SD} and T_{TD} represent the SD tracking threshold and the TD tracking threshold, respectively. Considering that a safe distance between vehicles will be kept when vehicles drive and that there is a relationship between this distance and the size of the vehicle, we set $T_{SD} = 1/3$ and $T_{TD} = 5$. If $SD(bb_i^k, tb_k)$ is greater than T_{SD} and $TD(bb_i^k, tb_k)$ is greater than T_{TD}, bb_i^k is determined to be a new vehicle.

Using this method to track vehicles has a low computing cost, especially compared with deep learning vehicle detection. The cost of this part is negligible, which does not affect the computational efficiency of vehicle counting.

4.4 The False Alarm Suppression Module Based on Bounding Box Size Statistics

For the problem of false vehicle detection, a false alarm suppression module based on the bounding box size statistics is added to the vehicle counting model. The false detection bounding box is usually larger or smaller than those of correct detections. Therefore, the false detection bounding box can be removed by comparing it with the size range of the correct detection. The process is as follows:

1) From the beginning of vehicle detection, the vehicle bounding box detected in each detection area is stored separately according to vehicle classification, and the height of every bounding box of each vehicle classification is averaged as $\bar{h}_k^c$, where c represents the vehicle classification and k represents the detection area.
2) Vehicles in each frame are detected, and the true detection value (TV) is used to determine whether a detected vehicle is a true detection. If it is a false detection, it is removed. The calculation is as follows:

$$TV = \begin{cases} 1, & \left|h_{bb_i^k} - \bar{h}_k^c\right| < T_{size} \times \bar{h}_k^c \\ 0, & otherwise \end{cases} \quad (4)$$

$$TV = \begin{cases} 1, & (1 + T_{size})\bar{h}_k^{car} < h_{bb_i^k} < (1 + T_{size})\bar{h}_k^{bus} \\ 0, & otherwise \end{cases} \quad (5)$$

where $h_{bb_i^k}$ represents the height of bounding box bb_i^k detected in area k and T_{size} is the threshold. Considering that vehicles of the same classification have different sizes, especially trucks, we use **Eq. 4** to determine a car and a bus, and use **Eq. 5** to determine a truck, setting the $T_{size} = 0.5$.

3) Step 1 and step 2 are executed cyclically, and $\bar{h}_k^c$ is updated until the end of the video.

The elimination of false detection may lead to missing detection, but it can be corrected by the abovementioned missing alarm suppression module based on vehicle tracking.

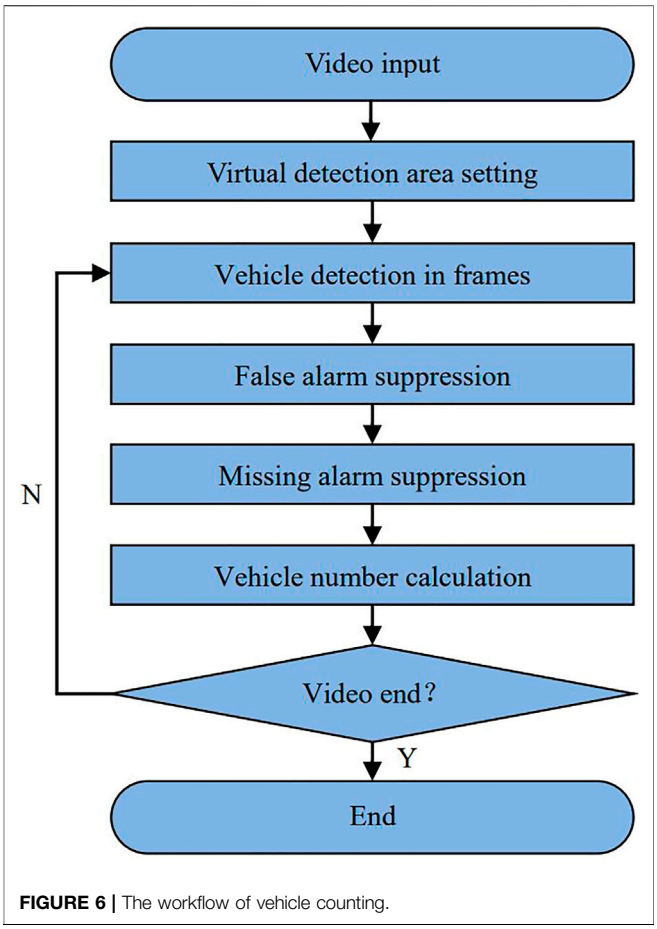

FIGURE 6 | The workflow of vehicle counting.

4.5 Vehicle Number Calculation

The process of vehicle counting based on fusing the virtual detection area and vehicle tracking is as follows (**Figure 6**):

1) Virtual detection area setting: according to the vehicle counting task for road sections or lanes, virtual detection areas are set up in each road section or lane.
2) Vehicle detection: Vehicles are detected frame-by-frame and the detected bounding boxes are taken as the output of the detection result, including classification, detected confidence, coordinate of the central point, width, and height.
3) False alarm suppression: In each detection area, according to the size of the detected bounding boxes, a detected bounding box is determined as false detection or not, and if so, false detection is eliminated.

TABLE 1 | Information on traffic videos.

Video	Light	Shooting direction	Traffic conditions	Resolution
1	Day	Front	Traffic lights	1,280 × 720
2	Day	Front	Light	1920 × 1,080
3	Day	Back	Heavy	1920 × 1,440
4	Night	Back	Heavy	1920 × 1,440
5	Day	Front	Traffic lights	1920 × 1,440
6	Night	Front	Traffic lights	1920 × 1,440
7	Night	Front	Traffic lights	1920 × 1,440
8	Night	Back	Light	1920 × 1,440
9	Day	Oblique front	Traffic lights	932 × 500
10	Day	Oblique back	Light	932 × 500

4) Missing alarm suppression: In each detection area, vehicle tracking based on the location of vehicle bounding boxes is carried out on the vehicles detected in the continuous frame sequence to determine whether the detected vehicle is a new vehicle.
5) Vehicle number calculation: Based on the virtual detection area and vehicle tracking, the flow curve of each vehicle type in every detection area is monitored to calculate the vehicle number.

5 EXPERIMENTS

To evaluate the performance of the proposed vehicle counting framework, four experiments are designed and carried out.

The experimental environment is a CPU: Intel Core i7-8700 3.20 GHz; Memory: 16 GB (2,666 MHz); GPU: NVIDIA GeForce GTX 1070, 8 GB.

The experimental data included ten traffic videos with different light conditions, shooting directions, traffic conditions, and resolutions (**Table 1**). All videos are captured for 5 min at 20 frames per second (fps). The MS COCO dataset is used as the basic training data because it has the characteristics of multiple small objects in a noncentral distribution in an image, which is more in line with the daily traffic scenario.

5.1 The Effectiveness of the Supplemental Dataset

Purpose: The purpose of the experiment is to evaluate the effectiveness of the supplemental dataset in the construction of the vehicle detection model.

The supplemental dataset is built as training data and testing data to further improve and evaluate the performance of the constructed vehicle detection model. According to the process introduced in **Section 3.3**, 1,660 images are refined annotated in total and then divided into training data and testing data; 1,000 for training and 660 for testing. In particular, only frame images in videos 1–8 are included, and videos 9 and 10 are used to evaluate the generalisability of the constructed vehicle counting model.

Based on vehicle data in the MS COCO dataset and the supplemental dataset, the following five vehicle detection models are trained, and their accuracies are evaluated. Model-coco is trained by 16,270 images with vehicles in the MS COCO dataset. Model-100, Model-250, Model-500 and Model-1000 are constructed as described in **Section 3.3**, and the number of supplemental training data points is 100, 250, 500 and 1,000, respectively. All models contain frame images of videos 1–8. The testing data for these five models are 660 images in the supplemental dataset, and the mean Average Precision (mAP) is used to represent the accuracy of these models. The mAP score is calculated by taking the mean AP over all classes and overall IoU thresholds, depending on different detection challenges that exist. In this study, AP for one object class is calculated for an Intersection over Union (IoU) threshold of 0.5. So the mAP@0.5 is averaged over all object classes. The accuracy of each model is shown in **Table 2**. As the amount of supplemental training data increases, the accuracy of the vehicle detection model is improved, especially from 48.78 to 73.07% with just 100 supplemental training data points. Thus, the supplemental dataset plays a role in vehicle detection model construction through transfer learning, which considerably improves vehicle detection accuracy by using only a small amount of supplemental data.

TABLE 2 | Vehicle detection accuracy of each model (mAP@0.5).

Model	Training data	All	Car	Bus	Truck
Model-coco	Coco	48.78	74.46	53.89	17.68
Model-100	coco +100	73.07	79.23	67.48	72.50
Model-250	coco +250	76.75	79.80	74.45	75.99
Model-500	coco +500	80.20	86.12	77.24	77.24
Model-1000	coco +1000	84.40	87.21	81.86	84.12

5.2 The Generalization of Vehicle Counting

Purpose: The purpose of the experiment is to evaluate the generalization of the vehicle counting model in videos with various light conditions, shooting directions, traffic conditions, and resolutions.

When employing Model-1000 constructed in Experiment 1 as the vehicle detection model, the vehicle numbers of the above 10 videos are counted, and the counting accuracy and computational efficiency are evaluated. **Figure 7** shows the screenshots of vehicle counting in traffic video. For direct-viewing expressions, only bounding boxes of vehicles passing through the detection area are displayed. First, in each video, virtual detection areas are set in each land. Then, vehicle counting is carried out in each detection area according to vehicle type, and the number of each type in each detection area is calculated as the counting results of the corresponding lane. Finally, the counting result of the road section is obtained by adding the counting results of all lanes. The counting results are shown in **Table 3**, where NR and ND refer to the real and detected number of vehicles, respectively, and CA (%) refers to the counting accuracy. The following conclusions can be drawn:

1) The model has strong performance in videos with various light conditions, shooting directions, traffic conditions, and resolutions, and the counting accuracy of the total number of vehicles and the number of three vehicle types all reach up to 99%. This shows that vehicle counting based on fusing virtual

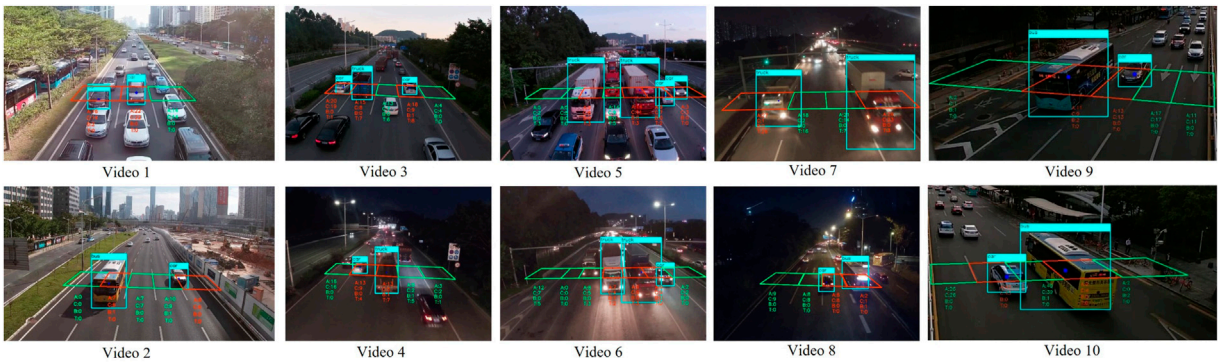

FIGURE 7 | Screenshots of vehicle counting in traffic videos.

TABLE 3 | Results of vehicle counting.

Video	Total			Car			Bus			Truck			Efficiency (fps)
	NR	ND	CA (%)	NR	ND	CA (%)	NR	ND	CA (%)	NR	ND	CA (%)	
1	145	145	100.00	141	141	100.00	4	4	100.00	0	0	100.00	22.2
2	226	226	100.00	216	216	100.00	10	10	100.00	0	0	100.00	21.3
3	99	100	98.99	67	68	98.51	2	2	100.00	30	30	100.00	20.2
4	136	136	100.00	83	82	98.80	3	3	100.00	50	51	98.00	20.2
5	127	127	100.00	100	100	100.00	1	1	100.00	26	26	100.00	20.2
6	138	139	99.28	109	109	100.00	2	2	100.00	27	28	96.30	20.2
7	119	117	98.32	55	54	98.18	0	0	100.00	64	63	98.44	20.2
8	254	254	100.00	247	247	100.00	7	7	100.00	0	0	100.00	20.2
9	151	151	100.00	144	144	100.00	7	7	100.00	0	0	100.00	22.5
10	184	184	100.00	171	171	100.00	13	13	100.00	0	0	100.00	22.5
Average	—	—	99.66	—	—	99.55	—	—	100.00	—	—	99.27	21.0

detection area and vehicle tracking can avoid the errors caused by missing detection and false detection, which further improves the accuracy of vehicle counting, although the accuracy of vehicle detection is not very high.

2) The counting accuracy of video 9 and video 10 also reaches 99%, which shows that the model has good generalisability.
3) The computational efficiency of the model is faster than 20 fps, which meets the requirement of real-time vehicle counting.

5.3 The Effectiveness of Transfer Learning

Purpose: The purpose of the experiment is to evaluate the effectiveness of transfer learning in the construction of a deep learning vehicle counting model.

The following three vehicle detection models are constructed during transfer learning.

1) Model-1: This model is a source model trained by vehicle data in the MS COCO dataset.
2) Model-2: This model is a target model initialised by parameters of Model-1 and trained by the merged datasets of vehicle data in the MS COCO dataset and the supplemental dataset.
3) Model-3: This model is a target model fine-tuned from Model-2 through further training on the supplemental dataset.

To verify the effectiveness of instance-based and parameter-based transfer learning, another two vehicle detection models are constructed:

4) Model-4: This model is a target model initialised by parameters of Model-1 and trained by just the supplemental dataset to validate the effectiveness of instance-based transfer learning.
5) Model-5: This model is a target model trained by the merged datasets of vehicle data in the MS COCO dataset and supplemental dataset without initialisation by the parameters of Model-1. The model is then fine-tuned by the supplemental dataset to validate the effectiveness of parameter-based transfer learning.

The above five models are used to count the vehicle number of 10 videos, and the counting accuracies of each model are combined and compressed. Because the final task of the vehicle detection model is to count vehicles and the vehicle counting method proposed in this paper can further improve the accuracy of vehicle counting when the accuracy of vehicle detection is not high enough, the accuracy of vehicle counting is used as the evaluation index of the accuracy of the model. The average counting accuracy of each model in 10 videos is shown in **Table 4**. From the experimental results, the following conclusions can be drawn:

TABLE 4 | Vehicle counting accuracy of each model (%).

Model	Total	Car	Bus	Truck	Average
Model-1	95.88	89.32	41.53	52.25	69.75
Model-2	99.32	99.42	85.67	86.60	92.75
Model-3	99.66	99.55	100.00	99.27	99.62
Model-4	99.39	99.16	97.57	98.29	98.60
Model-5	99.55	98.94	87.57	97.63	95.92

TABLE 5 | Vehicle detection accuracy of Model-1000-Tiny (mAP@0.5).

Model	Training data	All	Car	Bus	Truck
Model-1000-Tiny	coco +1000	71.21	72.70	75.20	65.73

1) Compared with other models, Model-3 has the best detection accuracy, and the counting accuracy of the total number of vehicles, as well as the number of three vehicle types all reach 99%, which shows that the vehicle counting model based on transfer learning performs well.
2) Model-1, Model-2, and Model-3 improve the counting accuracy step-by-step, which shows that transfer learning plays a role. Annotated data of the target task can be obtained from the source model, and further processing into refined annotated data improves the performance of the target model through transfer learning.
3) Compared with Model-4 and Model-5, Model-3 has better detection accuracy. It shows that instance-based and parameter-based transfer learning are both working. A more reliable deep learning model can be constructed without annotated data through transfer learning.

5.4 The Robustness of Vehicle Counting

Purpose: The purpose of the experiment is to evaluate the robustness of vehicle counting under the condition of a vehicle detection model with higher efficiency but lower accuracy.

In the proposed framework, the vehicle detection process takes most of the computing cost and depends on the efficiency of the basic model used in the vehicle detection model construction stage. The vehicle counting process avoids the errors caused by missing detection and false detection and further improves the accuracy of vehicle number calculation. Thus, the framework is flexible, and the trade-off between accuracy and efficiency can be made according to the requirement by choosing a deep learning object detection model with suitable performance in accuracy and efficiency as the basic model.

Tiny-YOLO is a simplified version of YOLO that has higher efficiency but lower accuracy. Tiny-YOLO is used as the basic model, and a vehicle detection model (Model-1000-Tiny) is constructed in the same way as Model-1000. Then, it is used to perform vehicle counting in 10 videos and evaluate the counting accuracy and computational efficiency. The vehicle detection accuracy of Model-1000-Tiny is shown in **Table 5**, which is more than 10% lower than Model-1000. The vehicle counting result is shown in **Table 6**; it can see that the counting accuracy of the total number of vehicles and the number of three vehicle types are relatively lower than that of Model-1000, but the gap is not as large as vehicle detection accuracy. It is higher than 98% in the total vehicle number and car number, 90.37% in bus number and 97.61% in truck number. Although the average counting accuracy of the bus number is not very high, it performs well in eight of the 10 videos. It only makes large mistakes in video 6 and video 8, while two buses are counted into one bus in video 6, and 7 buses are counted into five buses in video 8. However, this is likely caused by the small real number of buses; once there is an error, the accuracy is seriously reduced, and thus, overall, the model has good performance. However, in computational efficiency, using Tiny-YOLO as the basic model is more efficient than using YOLO, which can reach 53.4 fps, more than twice that of Model-1000. Thus, the proposed framework can maintain high accuracy of vehicle counting, although the accuracy of the vehicle detection model is not very high, and the trade-off between accuracy and efficiency can be made according to the requirements.

6 CONCLUSION AND FUTURE WORK

In this paper, a deep learning framework for video-based vehicle counting is proposed. The framework has two main

TABLE 6 | Results of vehicle counting (Tiny-YOLO).

Video	Total			Car			Bus			Truck			Efficiency (fps)
	NR	ND	CA (%)	NR	ND	CA (%)	NR	ND	CA (%)	NR	ND	CA (%)	
1	145	143	98.62	141	139	98.58	4	4	100.00	0	0	100.00	86.5
2	226	227	99.56	216	216	100.00	10	11	90.00	0	0	100.00	42.5
3	99	101	97.98	67	68	98.51	2	2	100.00	30	31	96.67	32.4
4	136	141	96.32	83	83	100.00	3	3	100.00	50	55	90.00	32.4
5	127	129	98.43	100	102	98.00	1	1	100.00	26	26	100.00	32.4
6	138	140	98.55	109	110	99.08	2	1	50.00	27	29	92.59	32.4
7	119	116	97.48	55	54	98.18	0	0	100.00	64	62	96.88	32.4
8	254	255	99.61	247	250	98.79	7	5	71.43	0	0	100.00	32.4
9	151	152	99.34	144	145	99.31	7	7	100.00	0	0	100.00	105.3
10	184	188	97.83	171	174	98.25	13	14	92.31	0	0	100.00	105.3
Average	—	—	98.37	—	—	98.87	—	—	90.37	—	—	97.61	53.4

tasks: deep learning vehicle detection model construction and vehicle counting. In deep learning vehicle detection model construction, to solve the problem of lacking annotated data, based on an open dataset, instance-based transfer learning and parameter-based transfer learning are adopted to construct a vehicle detection model with good performance. In vehicle counting, for the possible situation of vehicle missing detection and false detection, vehicle counting based on fusing virtual detection area and vehicle tracking is proposed. Missing alarm suppression module based on vehicle tracking and false alarm suppression module based on bounding box size statistics are designed to avoid vehicle counting errors caused by missing detection or false detection, which further improves the accuracy of vehicle counting. In this framework, the trade-off between accuracy and efficiency can be made according to the requirement by choosing a deep learning object detection model with a suitable performance in accuracy and efficiency as the basic model. Moreover, the proposed framework can improve the accuracy of vehicle counting although the accuracy of vehicle detection is not very high.

All the traffic videos used in this study are shot on straight roads. However, there are other scenarios in traffic surveillance, such as intersections and T-junctions. Although the model in this study has strong performance in straight road scenarios, making the model work well in different scenarios is an important problem to solve. Future work will consider scene adaptation to build a vehicle counting framework for different scenarios.

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

HL, ZY, BH, XK and XL: conceived the study, designed the vehicle detection and counting algorithms, conducted experiments, interpreted results, wrote and revised the article; RG: interpreted results, and revised the article.

FUNDING

We acknowledge the financial support by the National Natural Science Foundation of China (Project Nos. 71901147, 41901325, 41901329, 41971354, and 41971341), the Guangdong Science and Technology Strategic Innovation Fund (the Guangdong–Hong Kong-Macau Joint Laboratory Program, Project No. 2020B1212030009), the Research Program of Shenzhen S&T Innovation Committee (Project Nos. JCYJ20210324093600002 and JCYJ20210324093012033), the Science Foundation of Guangdong Province (Project Nos. 2121A1515012574 and 2019A1515010748), the Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, MNR, China (No. KF-2019-04-014), and Jilin Science and technology development plan project (No.20190303124SF).

REFERENCES

1. Zheng X, Cai Z. Privacy-preserved Data Sharing towards Multiple Parties in Industrial IoTs. *IEEE J Select Areas Commun* (2020) 38:968–79. doi:10.1109/jsac.2020.2980802
2. Cai Z, Zheng XS. Engineering, A Private and Efficient Mechanism for Data Uploading in Smart. *cyber-physical Syst* (2018) 7:766–75. doi:10.1109/TNSE.2018.2830307
3. Gao C, Fan Y, Jiang S, Deng Y, Liu J, Li X. Dynamic Robustness Analysis of a Two-Layer Rail Transit Network Model. *IEEE Transactions on Intelligent Transportation Systems* (2021). doi:10.1109/TITS.2021.3058185
4. Huang Q, Yang Y, Xu Y, Yang F, Yuan Z, Sun Y. Citywide Road-Network Traffic Monitoring Using Large-Scale mobile Signaling Data. *Neurocomputing* (2021) 444:136–46. doi:10.1016/j.neucom.2020.07.150
5. Huang Q, Yang Y, Xu Y, Wang E, Zhu KJWC, Computing M. Human Origin-Destination Flow Prediction Based on Large Scale Mobile Signal Data. *Wireless Communications and Mobile Computing* (2021). p. 2021. doi:10.1155/2021/1604268
6. Li Y, Li B, Tian B, Yao Q. Vehicle Detection Based on the and- or Graph for Congested Traffic Conditions. *IEEE Trans Intell Transport Syst* (2013) 14:984–93. doi:10.1109/tits.2013.2250501
7. Barcellos P, Bouvié C, Escouto FL, Scharcanski J. A Novel Video Based System for Detecting and Counting Vehicles at User-Defined Virtual Loops. *Expert Syst Appl* (2015) 42:1845–56. doi:10.1016/j.eswa.2014.09.045
8. Kamkar S, Safabakhsh R. Vehicle Detection, Counting and Classification in Various Conditions. *IET Intell Transport Syst* (2016) 10:406–13. doi:10.1049/iet-its.2015.0157
9. Mandellos NA, Keramitsoglou I, Kiranoudis CT. A Background Subtraction Algorithm for Detecting and Tracking Vehicles. *Expert Syst Appl* (2011) 38:1619–31. doi:10.1016/j.eswa.2010.07.083
10. Hofmann M, Tiefenbacher P, Rigoll G. *Background Segmentation with Feedback: The Pixel-Based Adaptive Segmenter*. Providence, RI: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (2012). p. 38–43. doi:10.1109/CVPRW.2012.6238925
11. Rosin PL, Ellis TJ. *Image Difference Threshold Strategies and Shadow Detection*. Citeseer: BMVC (1995). p. 347–56.
12. Kasturi R, Goldgof D, Soundararajan P, Manohar V, Garofolo J, Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol. *IEEE Trans Pattern Anal Mach Intell* (2009) 31:319–36. doi:10.1109/tpami.2008.57
13. Horn BK, Schunck BG. Determining Optical Flow. *Artif intelligence* (1981) 17:185–203. doi:10.1016/0004-3702(81)90024-2
14. Chen Z, Cao J, Tang Y, Tang L. Tracking of Moving Object Based on Optical Flow Detection. In: Proceedings of 2011 International Conference on Computer Science and Network Technology, Harbin, China, December 24–26, 2011. IEEE (2011). p. 1096–9. doi:10.1109/iccsnt.2011.6182151
15. Lange S, Ulbrich F, Goehring D. Online Vehicle Detection Using Deep Neural Networks and Lidar Based Preselected Image Patches. In: 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, June 19–22, 2016. IEEE (2016). p. 954–9. doi:10.1109/ivs.2016.7535503
16. Mundhenk TN, Konjevod G, Sakla WA, Boakye K. A Large Contextual Dataset for Classification, Detection and Counting of Cars with Deep Learning. In: European Conference on Computer Vision, Amsterdam, Netherlands, October 11–14, 2016. Springer (2016). p. 785–800. doi:10.1007/978-3-319-46487-9_48
17. Wang J, Zheng H, Huang Y, Ding X. Vehicle Type Recognition in Surveillance Images from Labeled Web-Nature Data Using Deep Transfer Learning. *IEEE Trans Intell Transportation Syst* (2017) 19:2913–22. doi:10.1109/TITS.2017.2765676

18. Suhao L, Jinzhao L, Guoquan L, Tong B, Huiqian W, Yu P. Vehicle Type Detection Based on Deep Learning in Traffic Scene. *Proced Comput Sci* (2018) 131:564–72. doi:10.1016/j.procs.2018.04.281
19. Krizhevsky A, Sutskever I, Hinton GE. Imagenet Classification with Deep Convolutional Neural Networks. *Adv Neural Inf Process Syst* (2012) 25–105. Available at https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.
20. Cao L, Wang C, Li J. Vehicle Detection from Highway Satellite Images via Transfer Learning. *Inf Sci* (2016) 366:177–87. doi:10.1016/j.ins.2016.01.004
21. Li X, Ye M, Liu Y, Zhu C. Adaptive Deep Convolutional Neural Networks for Scene-specific Object Detection. *IEEE Trans Circuits Syst Video Technol* (2017) 29 (9): 2538–2551. doi:10.1109/TCSVT.2017.2749620
22. Wang Y, Deng W, Liu Z, Wang J. Deep Learning-Based Vehicle Detection with Synthetic Image Data. *IET Intell Transport Syst* (2019) 13:1097–105. doi:10.1049/iet-its.2018.5365
23. Michalopoulos PG. Vehicle Detection Video through Image Processing: the Autoscope System. *IEEE Trans.veh.technol* (1991) 40:21–9. doi:10.1109/25.69968
24. Engel JI, Martin J, Barco R. A Low-Complexity Vision-Based System for Real-Time Traffic Monitoring. *IEEE Trans Intell Transportation Syst* (2017) 18:1279–88. doi:10.1109/tits.2016.2603069
25. Rosas-Arias L, Portillo-Portillo J, Hernandez-Suarez A, Olivares-Mercado J, Sanchez-Perez G, Toscano-Medina K, et al. Vehicle Counting in Video Sequences: An Incremental Subspace Learning Approach. *Sensors* (2019) 19:2848. doi:10.3390/s19132848
26. Badenas Carpio J, Sanchiz Martí JM, Pla F. Motion-based Segmentation and Region Tracking in Image Sequences. *Pattern Recognition* (2001) 34:661–70. doi:10.1016/s0031-3203(00)00014-5
27. Unzueta L, Nieto M, Cortes A, Barandiaran J, Sanchez P. Adaptive Multi-Cue Background Subtraction for Robust Vehicle Counting and Classification. *IEEE Trans Intell Transportation Syst* (2012) 13:527–40. doi:10.1109/tits.2011.2174358
28. Dai Z, Song H, Wang X, Fang Y, Yun X, Zhang Z, et al. Video-based Vehicle Counting Framework. *IEEE Access* (2019) 7:64460–70. doi:10.1109/access.2019.2914254
29. Girshick R, Donahue J, Darrell T, Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: IEEE Conference on Computer Vision & Pattern Recognition, Columbus, OH, June 23–28, 2014 (2014). p. 580–7.
30. Girshick R. Fast R-CNN. In: IEEE International Conference on Computer Vision, Santiago, Chile, December 13–16, 2015 (2015). p. 1440–8.
31. Ren S, He K, Girshick R, Sun J. Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. *Adv Neural Inf Process Syst* (2015) 28. Available at https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html.
32. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al. SSD: Single Shot Multibox Detector. In: European conference on computer vision, Amsterdam, Netherlands, October 11–14, 2016. Springer (2016). p. 21–37. doi:10.1007/978-3-319-46448-0_2
33. Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, June 27–30, 2016 (2016). p. 779–88. doi:10.1109/cvpr.2016.91
34. Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, July 22–25, 2017 (2017). p. 7263–71. doi:10.1109/cvpr.2017.690
35. Redmon J, Farhadi A. *Yolov3: An Incremental Improvement*. arXiv preprint arXiv:1804.02767 (2018).
36. Pan SJ, Yang Q. A Survey on Transfer Learning. *IEEE Trans knowledge Data Eng* (2009) 22:1345–59. doi:10.1109/TKDE.2009.191
37. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft Coco: Common Objects in Context. In: European conference on computer vision, Zurich, Switzerland, September 6–12, 2014. Springer (2014). p. 740–55. doi:10.1007/978-3-319-10602-1_48
38. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet Large Scale Visual Recognition Challenge. *Int J Comp Vis* (2015) 115:211–52. doi:10.1007/s11263-015-0816-y
39. Everingham M, Eslami SMA, Gool LV, Williams CKI, Winn J, Zisserman A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int J Comp Vis* (2015) 111:98–136. doi:10.1007/s11263-014-0733-5

Conflict of Interest: The 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.