Title page

Long-term monitoring system for full-bridge traffic load distribution on long-span bridges

Authors: Liangfu Ge[1,a], Danhui Dan[1,2,*], Ki Young Koo[3,b], Yifeng Chen[1,c]

Affiliations:

1 School of Civil Engineering, Tongji University, 1239 Siping Road, Shanghai, 200092, China;

2 Key Laboratory of Performance Evolution and Control for Engineering Structures of Ministry of Education, Tongji University, 1239 Siping Road, Shanghai, 200092, China;

3 Vibration Engineering Section, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QJ, UK.

*** Corresponding author, Professor. Email: dandanhui@tongji.edu.cn**

- ✓ **Mail Address:** *Room 711, Bridge Building, Tongji University, 1239 Siping Road, Shanghai, PR China;*
- ✓ **Mobile:** *86-13918075836*
- ✓ **Fax:** *86-21-55042363*

Coauthor email:

a: liangfu@tongji.edu.cn; b: k.y.koo@exeter.ac.uk; c: 1651798@tongji.edu.cn

Acknowledgement: This work is supported by the National key R&D Program of China (2017YFF0205605); the National Nature Science Foundation of China (Grant No. 5187849); Shanghai Urban Construction Design Research Institute Project ‘Bridge Safe Operation Big Data Acquisition Technology and Structure Monitoring System Research’; the Ministry of Transport Construction Science and Technology Project ‘Medium-Small Span Bridge Structure Network Level Safety Monitoring and Evaluation’; and the CSC (China Scholarship Council) Scholarship (202006260246).

Long-term monitoring system for full-bridge traffic load distribution on long-span bridges

Liangfu Ge[1], Danhui Dan[1,2,*], Ki Young Koo[3], Yifeng Chen[1]

1 School of Civil Engineering, Tongji University, 1239 Siping Road, Shanghai, 200092, China;
2 Key Laboratory of Performance Evolution and Control for Engineering Structures of Ministry of Education, Tongji University, 1239 Siping Road, Shanghai, 200092, China;
3 Vibration Engineering Section, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QJ, UK.

Abstract

Long-term monitoring of traffic loads across the whole bridge is of great significance for bridge health monitoring and safety assessment, especially for long-span and complex bridges. However, none of the existing schemes have operated long term, and the full-bridge traffic load (FBTL) monitoring has remained a concept. In this paper, an improved FBTL monitoring framework capable of long-term operation was proposed, through fusion of the weigh-in-motion system (WIMs) and multi-camera vision systems. In the system, vehicle detection and tracking algorithms with bettter accuracy and robustness were achieved with deep convolutional neural networks, which could effectively deal with the problem of target loss during long-term monitoring. A fast but accurate correction method for transverse vehicle position was proposed by applying projective geometry, ensuring the system could operate with a large number of target vehicles. The proposed FBTL monitoring system was successful in generating the traffic load distribution of a long-span cable-stayed bridge, demonstrating its technical feasibility and engineering applicability.

Key words: full-bridge traffic load, long-term monitoring, vehicle detection, vehicle tracking, fast vehicle position correction, long-span bridge

1 Introduction

Traffic load is one of the dominant live loads on bridges and the resulting structural response is a direct reflection of bridges' safety and operating condition. It is necessary to take traffic loads into consideration throughout the life-cycle of bridges, including bridge design, health monitoring, maintenance and even restrengthening. In particular, in the field of bridge health monitoring, an accurate measurement of FBTL is of great importance for challenging problems such as load effect assessment [1-2], model updating [3-4] and damage identification [5-6].

Due to the technical complexity of the direct FBTL measurements, three indirect approaches have been investigated.

(1) The first approach was using the inversion of structural responses, known as the BWIM technique [7-9]. It was shown that the BWIM method can estimate simple forms of loads (e.g, concentrated load) if an accurate baseline physical model is

available [8-9]. However, it is difficult to accurately determine the baseline model of a long-span bridge. Also, it is challenging to achieve load inversion mathematically for complex traffic situations with multiple vehicles (loads) at the same time.

(2) The second approach is using the pavement-based weigh-in-motion system (WIMs) with certain assumptions such as no overtaking and lane changing. Using WIMs data, Obrien et al. [10] and Zhou et al. [11] applied traffic flow theory to simulate the FBTL on different types of bridges. These studies verified the load simulation with the reasonable assumption that this would be useful for the long-term assessment of bridges. However, due to the difference from the actual situation, the FBTL simulation sometimes could not reproduce the real conditions of the bridges. For example, in the load analysis of the Yongjiang Bridge, Ge et al. [12] found that the WIMs data only would cause statistical underestimation of the traffic load effect.

(3) The third approach is to obtain complete traffic load information through a field load test, which uses controlled vehicles positioned in specific locations on the bridge. This method has been applied in many studies of model updating and been proven very effective for different types of bridge [13-15]. However, the load test inevitably requires an interruption of traffic, while providing a limited record of measurements due to the short test time.

These traffic load distributions from the indirect approaches are either quite different from the actual situation (especially for long-span bridges), or not enough to be used for long-term bridge health monitoring.

With the rapid development of sensors and computer vision technology, extensive studies have emerged in recent years for direct traffic load measurements, combining data from multiple types of sensors, of which the most common is the vision-based method [16-21]. The study of vision-based methods started with partial observation of traffic loads: Ryan and Al [16] applied the background difference method to identify the vehicle position on the bridge and presented a detailed discussion on the identification accuracy. Furthermore, adding information about vehicle weight, Chen et al.[17] employed the same method to obtain the vehicle position in mutiple localised areas on a cable-stayed bridge and estimated the FBTL distribution, under the assumption that the vehicles did not change lanes. To reproduce the traffic load distribution, Dan and Ge [18] started a new monitoring scheme for the complete FBTL observation by fusion of the WIMs and multiple cameras covering the whole bridge deck. Since then, some researchers have used deep learning techniques to improve this scheme. Zhou et al. [19] proposed a novel vehicle detection method based on the faster region-based convolutional neural network (Faster R-CNN), effectively improving the system robustness to ambient lighting. Furthermore, Ge et al. [20] proposed an improved method for better vehicle transverse positioning accuracy using the dual target detection and vehicle dimension estimation. This improvement showed good performance in terms of lighting robustness, accuracy and calculation speed.

Even though the framework of full-bridge traffic load (FBTL) monitoring has been improved significantly, there are still challenges for long-term FBTL monitoring systems, as outlined below.

(1) In real applications of the system, it is more convenient to install cameras on the side of the road. This introduces a large transverse positioning error, which can be accurately corrected by [20]. However, the correction takes a relatively longer time, especially for multi-lane long-span bridges.

(2) Occlusion is an obstacle for successful vehicle tracking, effectively detaching the weight information from the detected vehicle. A robust vehicle tracking method is required to ensure continuous vehicle tracking.

This paper proposes a long-term FBTL monitoring framework and improved methods for better performance and applicability. The contributions of this paper are:

(1) For the first time, FBTL monitoring has been realised on a long-span bridge, Yongjiang Bridge.

(2) A YOLO-v4 based vehicle detection model was shown to detect distant small vehicles, which was better for the real image dataset from Yongjiang Bridge.

(3) A robust vehicle tracking strategy is proposed to address the problem of vehicle occlusion, combining the vehicle motion with the appearance features.

(4) A fast correction method for transverse vehicle position is proposed, using regression on the correction factor.

This paper is organised as follows. Section 2 introduces the overall framework of the proposed FBTL monitoring system. Section 3 decribes the hardware implementation of the FBTL on Yongjiang Bridge. Sections 4-6 describe the improved deep learning vehicle detection model, the robust vehicle tracking method, and the fast position correction method, respectively. Section 7 describes the transformation from the image to the bridge coordiates. Section 8 shows the results of the proposed system to Yongjiang Bridge, and Section 9 presents the conclusion.

2 Overall framework for the long-term FBTL monitoring system

Full-bridge traffic load (FBTL) monitoring refers to the identification of the traffic load distribution on the whole bridge deck in terms of vehicle weight and position.

Figure 1 is the flowchart of the proposed long-term FBTL monitoring system, covering the six key steps, the corresponding preparations on hardware and software, and three core tasks, as introduced in detail in this paper.

The required inputs to the proposed system are the segments of WIM text data and videos from the same period of time, while the output is the spatial distribution of traffic loads on the whole bridge deck at the corresponding moment. In general, the videos contain the position information of all passing vehicles, while the WIM text data records the information of vehicle weight, speed, lane number and crossing time. According to the order of input to output, the execution steps of the long-term FBTL monitoring system are as follows.

Step 1: First frame detection and weight matching

The sensors required for the FBTL monitoring include two WIM systems and multiple high-definition (HD) cameras. The WIM systems are installed at both entrances of the bridge to obtain the information on entering vehicles, while the cameras are arranged along the bridge to record videos with overlapping fields of view (FOVs). Vehicles entering the bridge are recorded by both the WIM system and the first camera for each direction. The weight information from the WIMs is attached to the vehicle detected in the video until the vehicle leaves the bridge.

Step 2: Vehicle detection in each field of view (FOV)

For the video captured in each FOV, the vehicle detection is carried out frame by frame to get vehicle positions in the image coordinates. The core task of this step is to design a high-performance vision-based object detection model. To meet the requirements of long-term monitoring on long-span bridges, this model needs to be robust against environmental changes and capable of detecting small targets far from the camera.

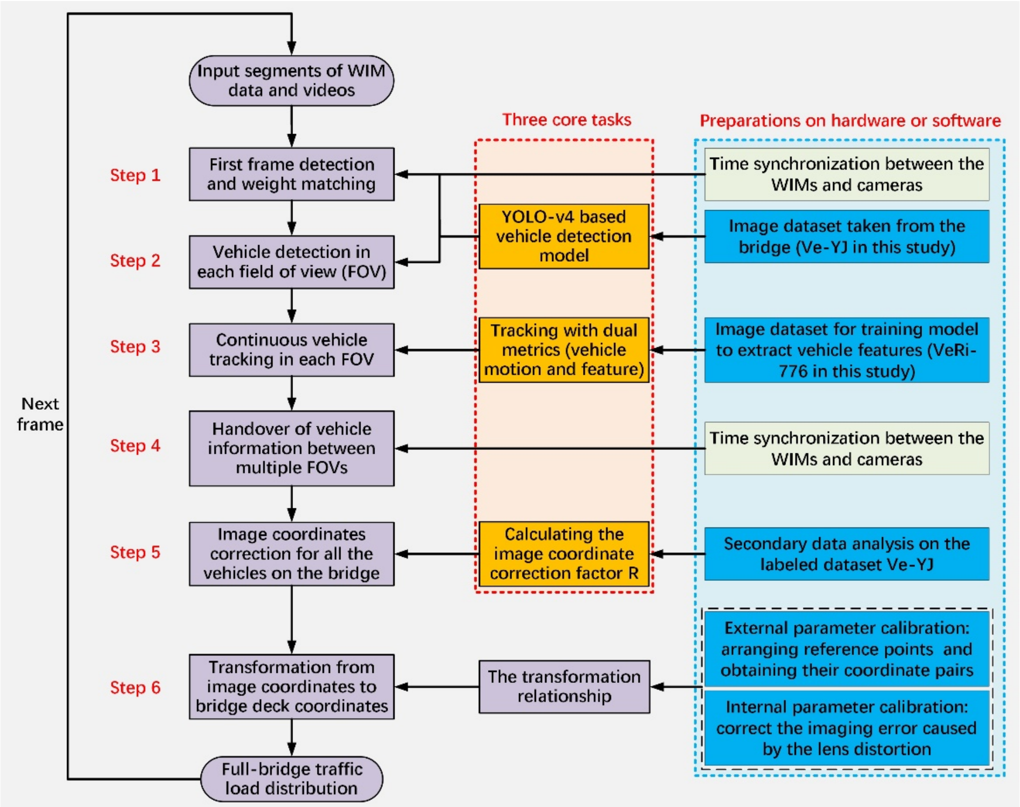

Input segments of WIM data and videos
Three core tasks
Preparations on hardware or software
Step 1
First frame detection and weight matching
Time synchronization between the WIMs and cameras
YOLO-v4 based vehicle detection model
Image dataset taken from the bridge (Ve-YJ in this study)
Step 2
Vehicle detection in each field of view (FOV)
Step 3
Continuous vehicle tracking in each FOV
Tracking with dual metrics (vehicle motion and feature)
Image dataset for training model to extract vehicle features (VeRi-776 in this study)
Next frame
Step 4
Handover of vehicle information between multiple FOVs
Time synchronization between the WIMs and cameras
Step 5
Image coordinates correction for all the vehicles on the bridge
Calculating the image coordinate correction factor R
Secondary data analysis on the labeled dataset Ve-YJ
External parameter calibration: arranging reference points and obtaining their coordinate pairs
Step 6
Transformation from image coordinates to bridge deck coordinates
The transformation relationship
Internal parameter calibration: correct the imaging error caused by the lens distortion
Full-bridge traffic load distribution

Figure 1. Flowchart of the long-term FBTL monitoring system

Step 3: Continuous vehicle tracking in each FOV

Vehicle tracking refers to the practice of continuously matching the detection results in adjacent frames to obtain vehicle trajectories over a period of time. This step inolves conducting continuous tracking of the vehicle in each FOV, while reducing trajectory loss, even if partial occlusion occurs. A novel dual-metric tracking method combining the vehicle motion and appearance features is proposed.

Step 4: Handover of vehicle information between multiple FOVs

To obtain the full-bridge traffic load distribution, it is necessary to establish the correspondence between vehicles captured by two adjacent cameras and ensure the transfer and sharing of vehicle information across the two FOVs. This step will realise it by applying the visibility of vehicles in different fields of view, i.e. the FOV line method proposed in [18].

Step 5：Correction of vehicle image coordinates

From a practical point of view, it is easier to install cameras on the side of the road along the bridge. However, this introduces a large error in the identification of vehicle transverse positions [20]. This step involves making a correction using a quick method presented in Section 6.

Step 6: Transformation from image coordinates to bridge deck coordinates

For each camera, the transformation relationship from the image coordinates to bridge deck coordinates is different. Different transformation matrices of the cameras are calculated through the projective geometry method [18] and applied to convert all image coordinates into the bridge coordinates.

After the above steps, the FBTL monitoring system can provide the full-bridge traffic load distribution at the current frame. For the next calculation frame, repeat the above 6 steps until the calculation of input segements is done. Throughout the process, vehicle detection, tracking and position correction are three key tasks. To complete these tasks properly, some preliminary preparations are also required, such as time synchronisation, image dataset making and camera parameter calibration. In the following sections, these key tasks and preparations are discussed for the FBTL monitoring system developed using Yongjiang Bridge, Ningbo, China.

3 Hardware of the FBTL monitoring system on Yongjiang Bridge

In this study, the long-term FBTL monitoring system is developed using Yongjiang Bridge, a single-tower cable-stayed bridge, which is about 200 meters in length. This bridge carries heavy urban traffic, equipped with a complete structural health monitoring system, including sensors such as WIMs (2), cameras (4), GPS (1), accelerometers (22), displacement gauges (4) and strain gauges (16), (Figure 2.)

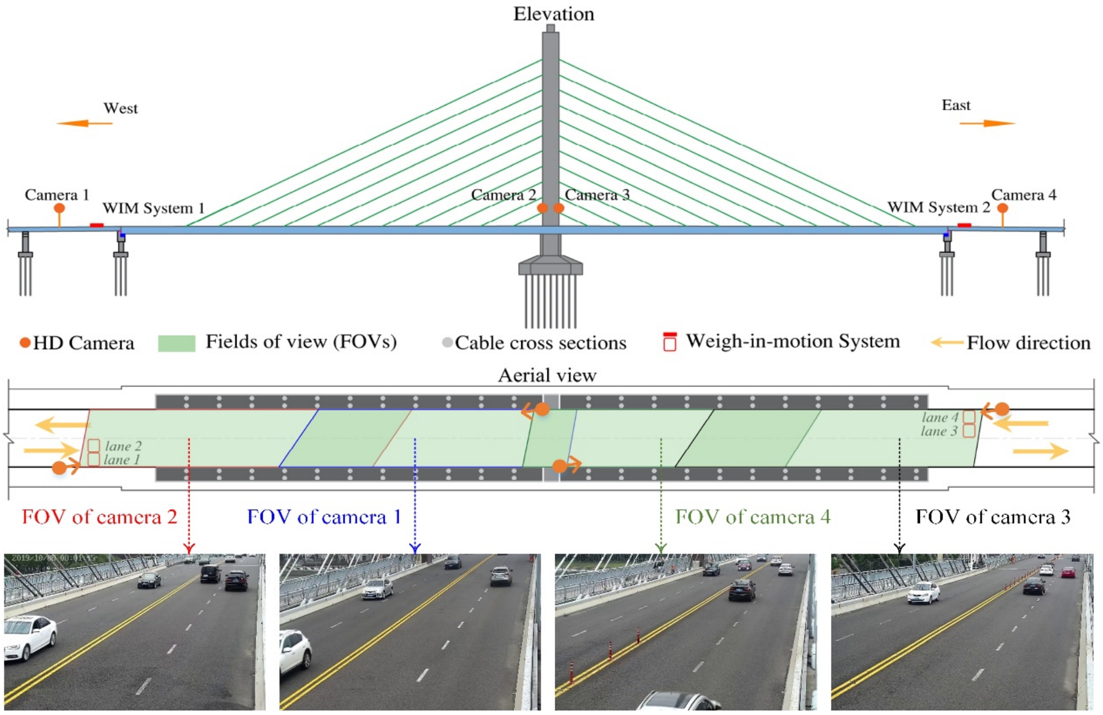
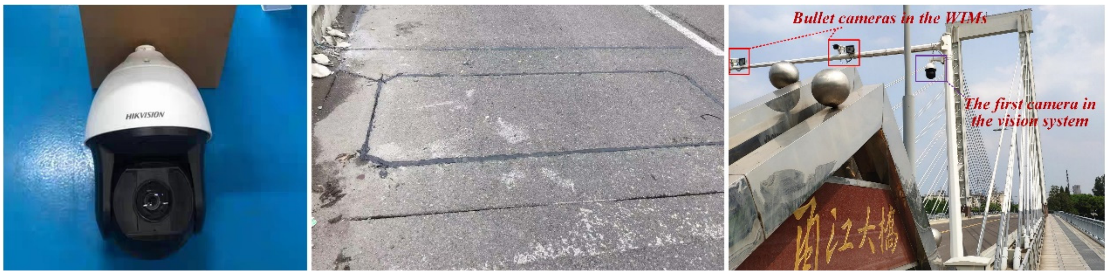


Figure 2. The FBTL monitoring hardware layout on Yongjiang Bridge

Figure 3 shows the HD cameras and WIM systems. Two WIM systems (MEAS quratz sensor of 1.75 m length) were arranged at both the entrances of the bridge, and four HD cameras (Hikvision HD-Camera, DS-2DF8234 4M pixel) were installed along the bridge with overlapping fields of view (FOVs). The WIM system equipped each lane with an additioanl camera for recognising vehicle registration numbers.


HIKVISION
Bullet cameras in the WIMs
The first camera in the vision system

(a) Hikvision HD-Camera (b) MEAS Quartz WIM Sensor (c) Installed cameras on the bridge

Figure 3. Details of the HD cameras and WIM system

For the FBTL monitoring, the WIMs could obtain information on each vehicle arriving at the bridge, including its weight, speed, arrival time, lane number, and registration number. Meanwhile, four cameras provided FOVs covering the whole bridge deck (shaded in Fig.2) and recorded videos of the vehicle. These videos were used to identify the vehicle position on the bridge by using the vehicle detection and tracking algorithms. Then, all the videos obtained by the cameras and the text data from WIMs were transferred over an optical fiber network to the server privately owned by the Ningbo Municipal Administration.

The cameras and WIM systems have been generating videos and WIM text files respectively. Due to the limitations of the server, only the latest 2 months of videos are available to download. Data processing is manually completed on downloaded video and WIMs files. It should be noted that, to fuse the information from the two different sensors, they must be time synchronised. Time synchronisation by GPS time alignment [21] is a promising approach. However, in this study, time synchronisation was carried out manually, using vehicle registration numbers appearing on the videos and the WIMs text files.

4 Optimised vehicle detection model using YOLO-v4

The first core task of the FBTL monitoring system is to detect vehicles in the video with the aid of computer vision technology, that is, to mark the position of vehicles with bounding boxes in each frame of the video sequences. In fact, the identification of vehicle position has attracted extensive attention, and many relevant studies have been carried out in the field of bridge health monitrong. From the initial application of traditional image processing methods [17,18] to the incorporation of convolutional neural networks [20, 25], the latest developments have been able to accurately identify most vehicles, except for the targets far away from the camera, which present very small pixel clusters in the image. However, for aesthetics and economic considerations, it is common to reduce a number of cameras on the bridge, expanding the monitoring FOV, resulting in problems with small target detection. This section introduces a novel detection model based on YOLO-v4 to address the small target detection problem.

4.1 Image dataset Ve-YJ for model training

For training a deep learning model for bridge monitoring, a custom image dataset named Ve-YJ was created, as opposed to using general datasets such as MS COCO and ImageNet.

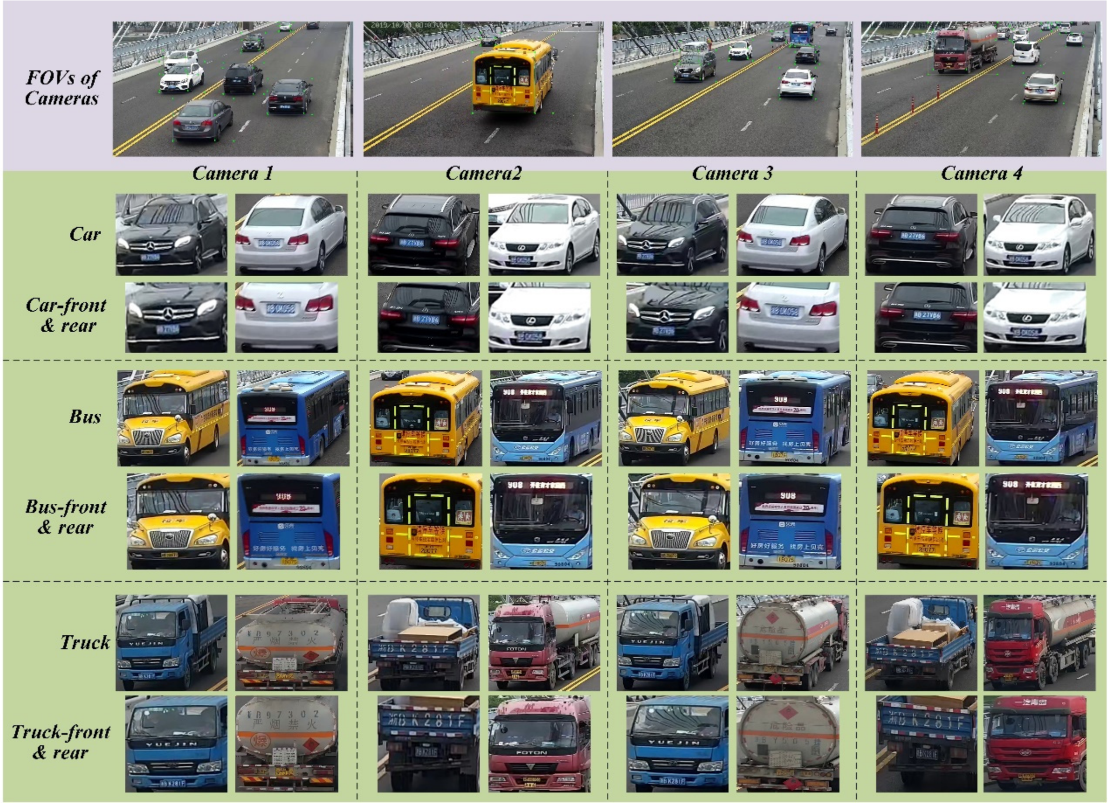


Figure 4. The composition of the Ve-YJ image dataset

Ve-YJ consists of a total of 24,348 pictures, with a resolution of 1,920×1,080, randomly selected from 3 days of monitoring video taken from 5am to 7pm. The pictures were chosen from different levels of traffic and light conditions, and labeled for three types of vehicles distributed in different positions of the camera FOV (Fig.4). The three classes of vehicles (cars, buses and trucks) were annotated with bounding boxes to record their positions. Inspired by literature [20], when the vehicle is fully visible in the FOV, its front and rear were also labeled for the correction of transverse vehicle positions on the bridge. Therefore, a total of 9 classes of objects were annotated, i.e, car, car-front, car-rear, bus, bus-front, bus-rear, truck, truck-front and truck-rear. The number and proportion of different classes in the Ve-YJ are shown in Table 1.

Table 1. The number and proportion of different classes in the image dataset Ve-YJ

Class	car	car-front	car-rear	bus	bus-front	bus-rear	truck	truck-front	truck-rear
Quantity	52706	14937	16764	1245	649	790	547	268	263
Proportion	59.78%	16.94%	19.01%	1.41%	0.74%	0.90%	0.62%	0.30%	0.30%

The Ve-YJ dataset was used not only as training and test datasets for the YOLO-v4-based vehicle detection model, but also a dataset for the transverse vehicle position correction detailed in Section 6.

4.2 YOLO-v4

The YOLO (You Only Look Once) [26] network is a single-stage CNN-based object detection algorithm, which directly calculates a classification and position coordinates of each target on the input image. YOLO-v4, which was proposed in 2020, integrates the advantages of various types of target detection networks and achieves the best trade off between accuracy and speed (Figure 5). The main backbone network used for training and extracting image features is CSPDarknet53, an open source neural network framework with an improved performance. Path Aggregation Network (PANet) was used as the neck network to fuse extracted features from different detector levels, instead of FPN in YOLO-v3 [27]. But the head of YOLO-v3 was still retained in YOLO-v4.

In the structure of our vehicle detection model, the composition and fuctions of main modules are as follows.

- The CBL (Figure 5) represents a module composed of a convolution layer, a batch normalisation layer and a Leaky-ReLU activation function, which was the most frequently used in the YOLO-v4 for feature extraction.
- The CBM module is also used for feature extraction, like the CBL, but with the activation function of MISH instead of Leaky-ReLU.
- The SPP, a spatial pyramid pooling layer, which mainly transforms convolution features of different sizes into pooled features with the same length [28].
- The CSP refers to the Center and Scale Prediction module that could enhance the learning ability of CNNs by dividing low-level features into two parts and then fusing cross-level features [29].

Moreover, the YOLO-v4 based vehicle detection model used CIOU [30] as the loss function, which considers the overlap area, centroid distance and aspect ratio of the predicted and ground truth bounding boxes, greatly improving the calculation accuracy of the vehicle position in the image.

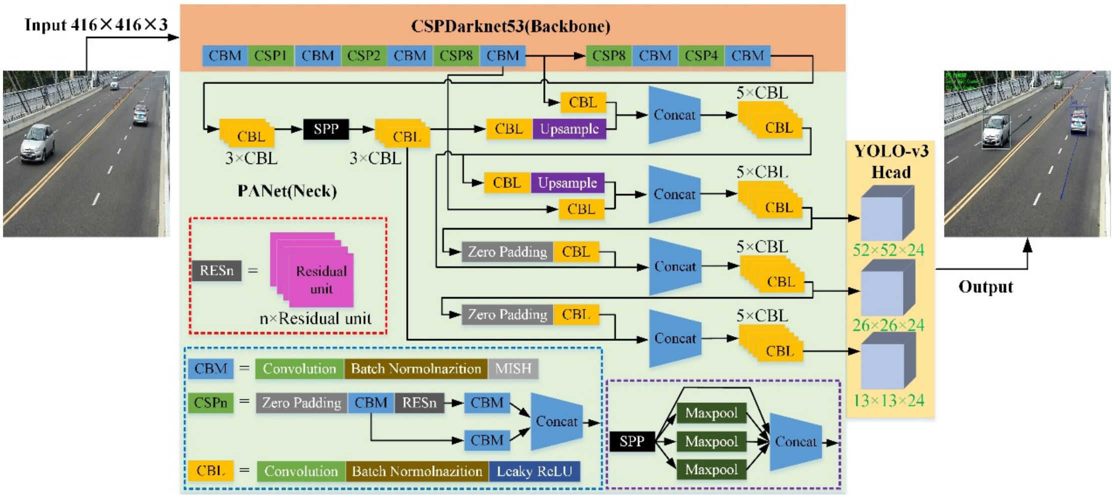


Figure 5. The architecture of the vehicle detection model based on YOLO-v4

The training process of this detection model involves undertaking the following steps.

Step 1: Prepare data. The image dataset Ve-YJ was employed in this study. The labeled 24,348 images were divided into a training set comprising 19,478 images (80%) and a test set with 4,870 images (20%).

Step 2: Data augmentation. Before being fed into the network, all images in the training set were scaled down to the resolution of 416×416 and processed, using a variety of data augmentation techniques, including rotation, mirroring, colour and brightness adjustment, and blurring.

Step 3: Model training and testing. Referring to official weights of the CSPDarknet53 network pretrained from the COCO dataset, the proposed vehicle detection model was trained and tested on a computer configured with GPU (NVIDA RTX 2070) and CPU (Intel Core i7-6700@3.4GHz). The training performed 20,000 iterations, and the key parameters are set to batch size = 8 and initial learning rate = 0.001.

After training, a test is conducted on the test dataset to verify the detection accuracy for different types of targets in comparison to a YOLO-v3 based model [19] as shown in Fig.6.

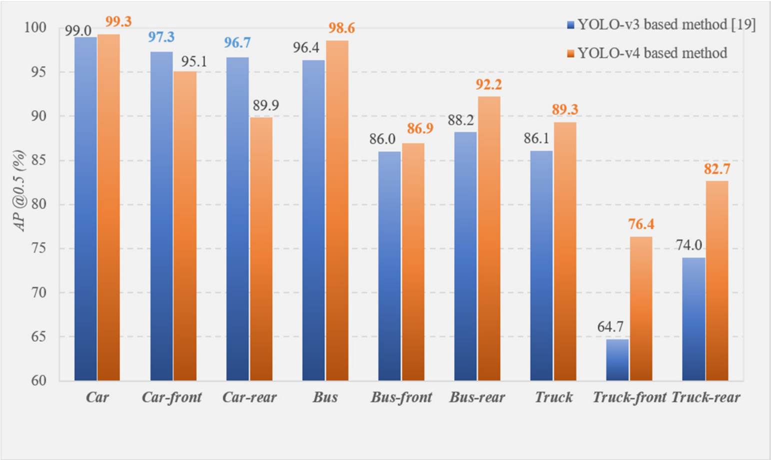


Figure 6. Performance comparision of our method and YOLO-v3 based method

In Fig.6, *AP@0.5* represents the average precision of the model for an IOU threshold of 0.5. The result showed that the proposed model outperforms the YOLO-v3 based model on detection accuracy for almost all types of targets, except car-front and car-rear. The detection accuracy of cars was much higher than that of buses and trucks. Such test results may be explained by the following two reasons:

(1) The front and rear profiles of the cars in the distant area of FOV were not annotated in the dataset Ve-YJ, because the transverse position error is negligible for distant cars. The YOLO-v4 based model is more sensitive to small targets, so it can still detect the unannotated car fronts and rears in the area far away from the camera, leading to a lower *AP* for these two classes.

(2) In the Ve-YJ, the number of buses and trucks in the image was much lower than that of cars, resulting in lower detection accuracy of them. To improve accuracy, it is recommended that the dataset is further expanded or another published

vehicle image dataset is integrated, to increase the samples of buses and trucks.

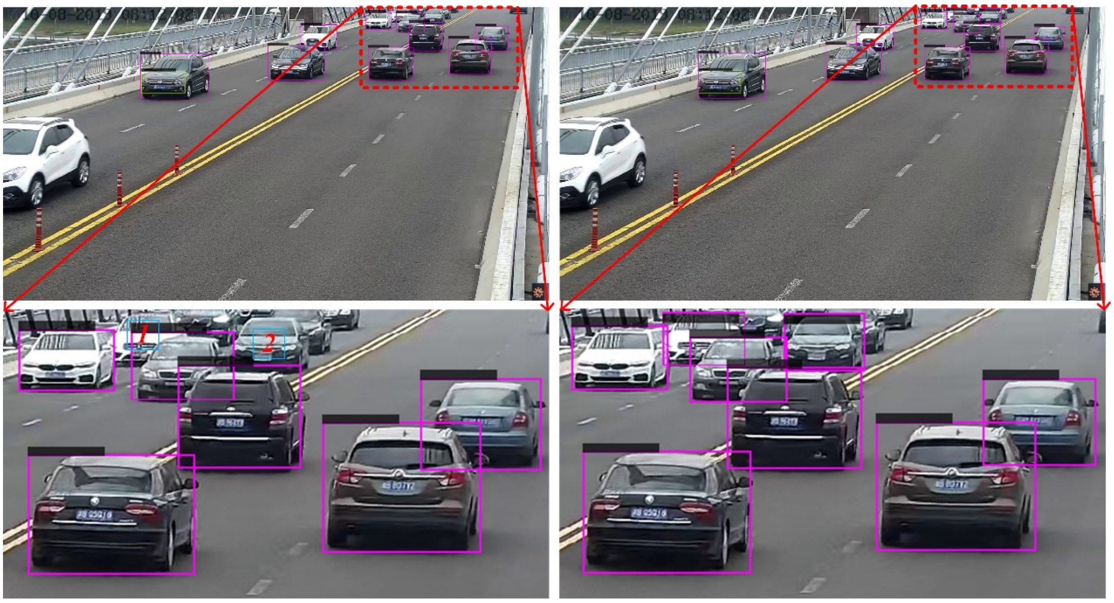


Figure 7. The ability of different methods to detect small targets
(left: YOLO-v3 based method[19]; right: YOLO-v4 in this study)

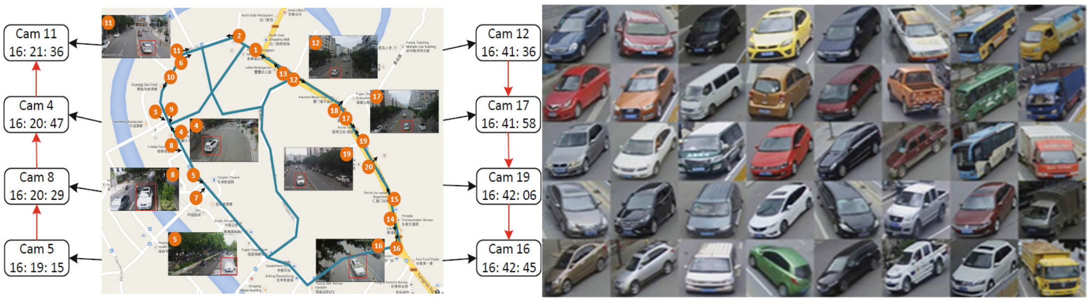
To further verify the advantage of the proposed model, the two methods were applied to a new scene never before used in training, as shown in Fig.7, with the detection results of multiple targets at the far end of the FOV (the red dashed rectangle). The result shows that the proposed method demonstrated better performance in detecting small vehicles and obtaining more compact bounding boxes, even if both the two methods show a robust performance when applied to the new monitoring scene

5 Vehicle tracking by combining Kalman filter and apparent feature

According to the authors' previous studies [18,20], Kalman filtering based vehicle tracking achieved a satisfactory result in short-term tracking tasks. However, with a longer tracking time, occlusion between the vehicles in different lanes may occur (see Fig.9), which leads to losing tracked trajetories. A robust tracking algorithm combining Kalman filter and appearance features is discussed in this section.

5.1 Image dataset for model training

It is necessary to prepare an image dataset to train a deep learning model for extracting vehicle appearance features. Instead of making our own dataset, the dataset named VeRi-776 [23] was used for the model training, with permission from the authors. As shown in Fig.8, VeRi-776 contains over 50,000 images of 776 vehicles, captured by 20 cameras on a one km^2 road network. Each vehicle was annotated with a bounding box, colour, type, the IDs of vehicle and camera, helping to train the model to distinguish the same car from different views.



(a) (b)

Figure 8. (a) the spatiotemporal information of a vehicle in the surveillance network; (b) vehicle images captured in different viewpoints. (cited from [22])

5.2 Kalman filter and appearance features

In the long-term tracking scenario, it is difficult to retrieve a lost trajectory when only relying on the prediction of vehicle motion information. A new robust tracking system with dual metrics is proposed.

The first metric is based on the detected vehicle bounding box and predicted ones using the Kalman filter with constant velcity motion and a linear observation model. In the tracking scenario, the state vector and measurement vector of the Kalman filter is defined as $(x,y,\gamma,h,\dot{x},\dot{y},\dot{\gamma},\dot{h})$ and (x,y,γ,h) respectively, where (x,y) are the bounding box center, γ is aspect ratio, h is height and $(\dot{x},\dot{y},\dot{\gamma},\dot{h})$ are their respective velocities in the image coordinates. Using the YOLO-v4 model, it is easy to obtain the measurement vector (x,y,γ,h) and the predicted Kalman state (x^p,y^p,γ^p,h^p) in the next frame.

Essentially, using the Kalman filter vehicle tracking is a matching problem between vectors (x,y,γ,h) and (x^p,y^p,γ^p,h^p) in each image frame, followed by forming a trajectory for successful matches over multiple frames. The solution of the matching problem and the rules for trajectory generation need to be explained.

For the matching problem, the Mahalanobis distance was used between the predicted Kalman states and newly arrived measurements on each image frame:

$$d_1(i,j)=(\vec{d}_j-\vec{p}_i)^T\mathbf{S}_i^{-1}(\vec{d}_j-\vec{p}_i) \qquad (1)$$

where $\vec{p}_i$ denotes the predicted bounding box of the i-th track, $\mathbf{S}_i$ the covariance matrix of four dimensional variables, and $\vec{d}_j$ the measurment vector of the j-th bounding box detection. The smaller $d_1(i,j)$ means a higher degree of matching.

With a threshold value $d_1(i,j)<d_1^{Threshold}$, the newly arrived measurement can be

matched with an existing trajectory. However, in case matching cannot be achieved, the exisiting trajectory stops and new detections may appear. This situation needs to be rectified. For the former, starting from the last successful measurement, the number of frames where track *k* have been continously exisited is counted as age a_k , which is incremented during the Kalman filter prediction and set to 0 after succesful matching. The maximum track age is defined as A_{max}. Tracks that exceed A_{max} are deleted, because they are considered to have left the scene. For the latter situation, a detection is assigned to a new trajectory only if there is no matching trajectory for consecutive b_k frames (b_k = 5 in this study).

The second metric is the similarity of vehicle appearance features, which was also integrated to the matching problem in this study. For the *j*-th bounding box detection, an appearance feature vector $\vec{f}_j$ is computed and normalised $\left\|\vec{f}_j\right\| = 1$. Furthermore, a feature gallery $\boldsymbol{F}_k = \{\vec{f}_k^{\,i}\}_{i=1}^{N_k}$ of the last $N_k = 100$ frames is kept for each track *k*. Then, the second metric to measure the similarity between the *i*-th track and *j*-th detection is defined as:

$$d_2(i,j) = \min\{1 - \vec{f}_j^{\,T}\vec{f}_k^{\,i} \mid \vec{f}_k^{\,i} \in \boldsymbol{F}_k\} \tag{2}$$

In practice, there are many methods that can be selected for the extraction of image features in the bounding box, including the traditional methods such as SIFT [31] and HOG [32] and deep learning methods. Herein, CNN architecture that has used for person re-identification[33-34] was chosen. This CNN is relatively shallow, to allow fast training and inference, as shown in Table 2. All the input images are rescaled to 128×64 and presented in RGB colour space. After a series of convolution operations, a 128-dimensional feature vector is extracted by *Dense* 10 and then projected onto the unit hypersphere by the l_2 normalisation to output the appearance feature vector $\vec{f}_j$.

The image datasets VeRi-776 and Ve-YJ were employed in the training process of the CNN for further performance improvement.

Table 2. Overview of the CNN architecture for vehicle appearance feature extraction

Name	Patch Size/Stride	Output Size
Conv 1	3×3/1	32×128×64
Conv 2	3×3/1	32×128×64
Max Pool 3	3×3/2	32×64×32
Residual 4	3×3/1	32×64×32
Residual 5	3×3/1	32×64×32
Residual 6	3×3/2	64×32×16
Residual 7	3×3/1	64×32×16
Residual 8	3×3/2	128×16×8
Residual 9	3×3/1	128×16×8

Dense 10	128
l_2 normalisation	128

Figure 9 shows the typical tracking process of an occluded vehicle based on the metric d_1 only (Fig.9 left) and a combination of the two metrics (Fig.9 right), where 3 different image frames are used for illustration. In frame 621, it was observed that the grey car (No. 8) was about to change lanes and overtake the white car (No. 6). Then, in the 627 *th* frame, Car 8 changed lanes and severely occluded Car 6, making Car 6 undetectable. In this case, our goal is to recapture Car 6 after occlusion. After losing track for dozens of frames, Car 6 was detected again at the 640 *th* frame. As shown in Fig.9, the photos at frame 640 presented the re-identification results of Car 6 using the two methods. Obviously, the only d_1-based method renumbers the white car as 11, losing the trajectory of Car 6. However, even if the trajectory of Car 6 was lost for more than ten frames, the proposed tracking scheme recovered its identity and achieved continuous tracking.

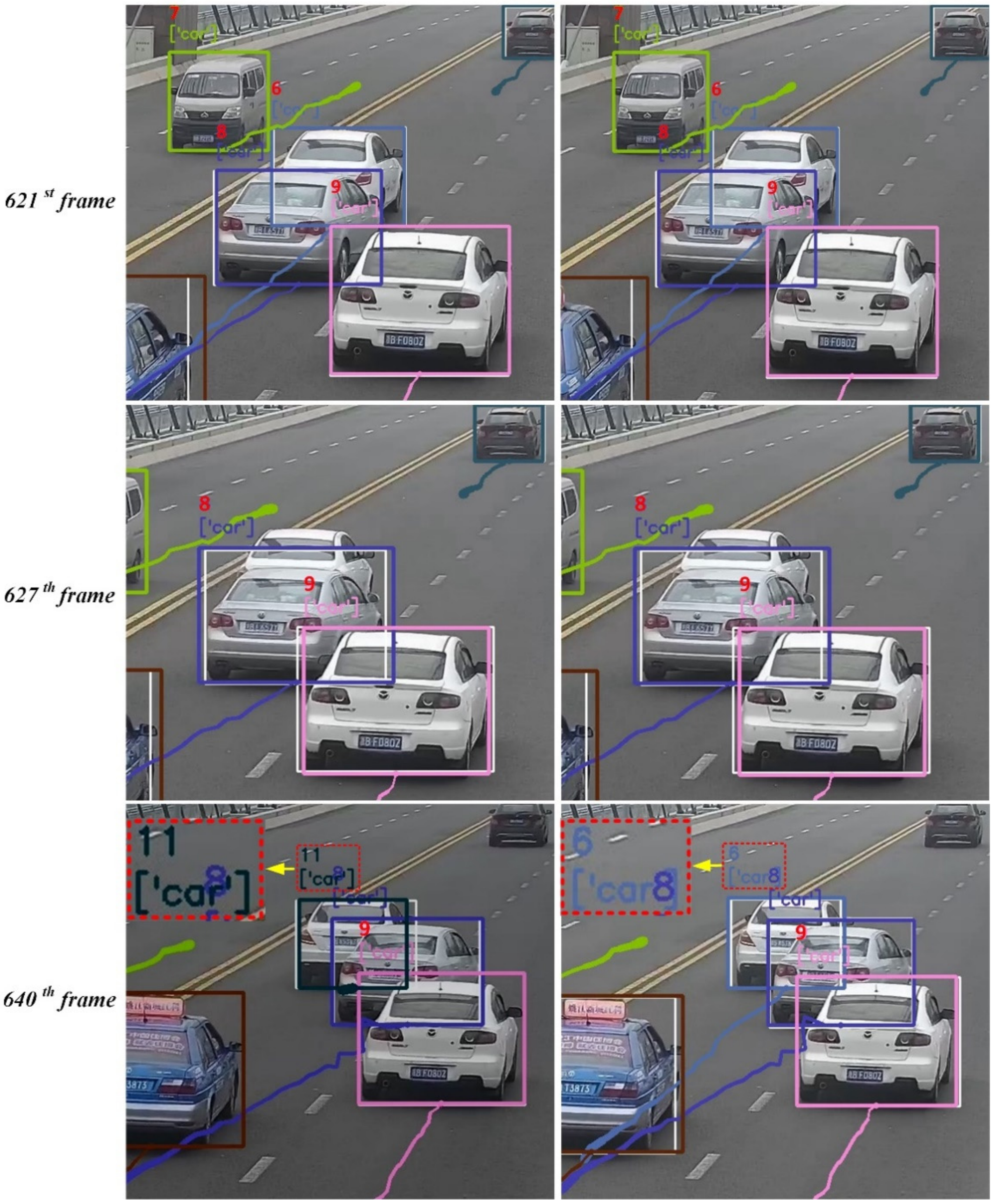


Figure 9. Tracking results when encountering occlusion (left: only d_1-based method, right: the proposed method)

It should be noted that the Kalman filter method may continue to predict vehicle position without measurement. In theory, by adjusting the maximum track age A_{max}, the predicted data can be retained and used to reidentify the lost trajectory. To investigate this possibility, the case in Fig.9 was considered. Figure10 showed the photo at the *640 th* frame when Car 6 was detected. The detected bounding box of Car 6, the predicted bounding boxes of Car 6 and Car 8, and the Mahalanobis distance d_1 were marked in Fig.10. It was seen that, after a dozen frames of trajectory loss, the predicted result of Car 6 deviated so much that the d_1 from the detected bounding box of Car 6 to the predicted bounding box of Car 8 was smaller than that of its own predicted bounding box. This demonstrated that using the metric d_1 alone made it difficult to retrieve a lost trajectory, while the proposed dual-metric tracking method effectively overcame the occlusion problem.

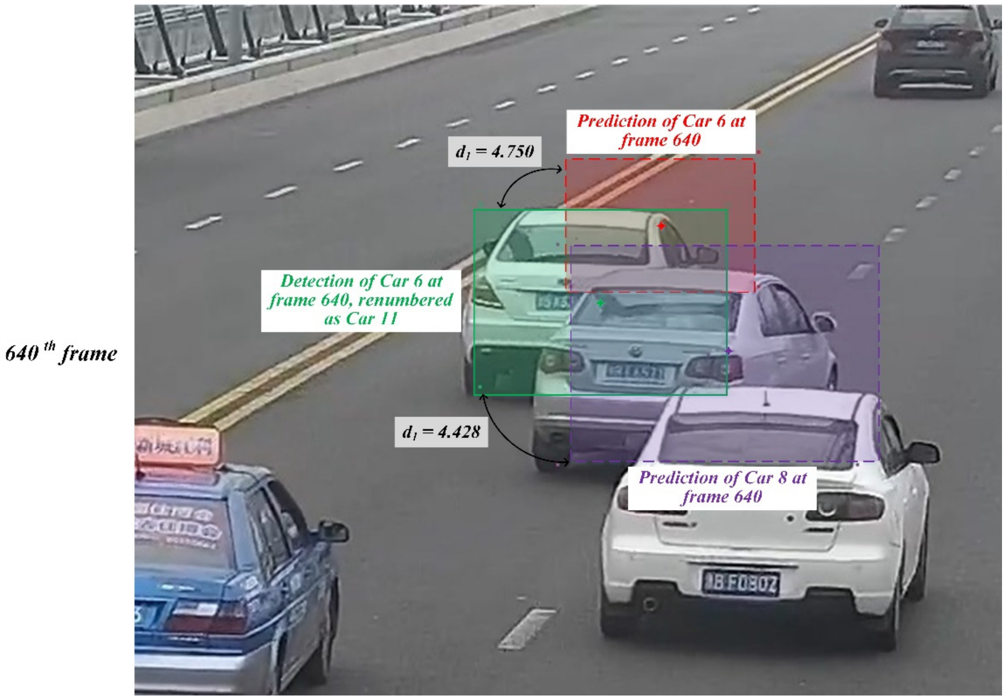

640 th frame
Prediction of Car 6 at frame 640
d1 = 4.750
Detection of Car 6 at frame 640, renumbered as Car 11
d1 = 4.428
Prediction of Car 8 at frame 640

Figure 10. Detected and predicted bounding boxes of Car 6 and Car 8 at frame 640

6 Fast vehicle transverse position correction using regression

By training the model based on the YOLO-v4, obtaining the bounding boxes of the vehicle and its front and rear parts was achievable. Based on the bounding boxes of the detected vehicles, vehicle fronts and rears, an accurate position correction also proved accessible in our previous study [20]. An efficient correction method is proposed using whole vehicle detection and a pre-determined mathmatical relationship, rather than time-consuming dual bounding boxes detection.

Figure 11 presents part of an image taken by Camera 3 on the Yongjiang Bridge. The red and purple dashed lines marked the sides of the bounding boxes, and the central axis of the front and rear profiles were indicated by yellow dashed lines, whose corresponding image coordinate x_{center} is the vehicle position to calculate. Based on the image coordinates presented in Fig.11, the correction factor R is defined as:

$$R_j = \frac{x_3^j - x_2^j}{x_3^j - x_1^j} \tag{3}$$

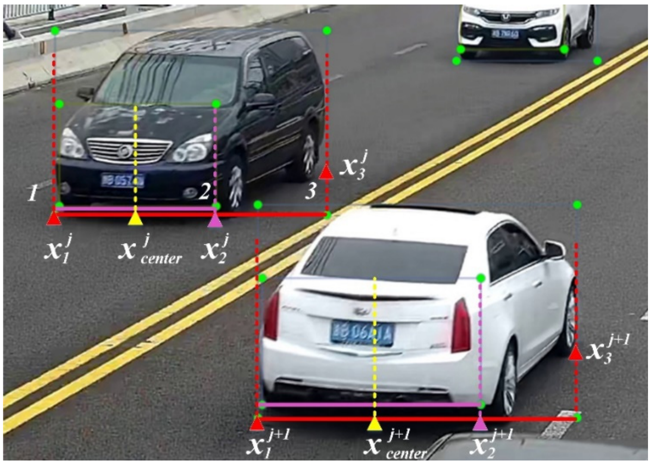


Figure 11. Key points of correction factor calculation in the image

Assuming that the camera shooting direction is parallel to the direction of the traffic flow, the relationship between the image coordinate x and the bridge deck coordinates (X, Y, Z) was derived based on the projective geometry [20].

$$x = \frac{-f(X - D_x)}{\sin\alpha(Y - D_y) + \cos\alpha(Z - D_z)} \tag{4}$$

where f is the camera focal length, and α is the shooting angle of the camera (α=0 when shooting vertically upward). D_x, D_y and D_z are the distances from the origin of the bridge deck coordinate system to the camera in the three directions, respectively. Since all the key points in Fig.11 (marked as triangles) are on the pavement, Z is set to 0 here. The image coordinates in Eq.(3) can be expressed as:

$$\begin{cases} x_1^j = \dfrac{-f(X_1^j - D_x)}{\sin\alpha(Y_1^j - D_y) - \cos\alpha D_z} \\ x_2^j = \dfrac{-f(X_1^j + W_j - D_x)}{\sin\alpha(Y_1^j - D_y) - \cos\alpha D_z} \\ x_3^j = \dfrac{-f(X_1^j + W_j - D_x)}{\sin\alpha(Y_1^j + L_j - D_y) - \cos\alpha D_z} \end{cases} \tag{5}$$

where W_j and L_j refer the width and length of the j-th vehicle. By substituting Eq.(5) into Eq.(3)

$$R_j = \frac{L_j(W_j + X_1^j)\sin\alpha}{D_z W_j \cos\alpha + (L_j X_1^j - W_j Y_1^j)\sin\alpha} \tag{6}$$

According to the transformation procedure introduced in [18], all the image coordinates (x, y) can be converted to bridge deck coordinates (X, Y, Z) as follows

$$Z_{cj}\begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{bmatrix} \tag{7}$$

In this equation, Z_{cj} denotes the distance from the spatial point j (point 1 in Fig.11) to the camera optical centre, and **M** is the transformation matrix. Eq.(6) can be rewriten as

$$R_j = \frac{Z_{cj}\tilde{m}_{11}x_1^j + Z_{cj}\tilde{m}_{12}y_1^j + Z_{cj}\tilde{m}_{13} + W_j}{Z_{cj}(\tilde{m}_{11}x_1^j + \tilde{m}_{12}y_1^j + \tilde{m}_{13}) + \left[Z_{cj}(\tilde{m}_{21} + \tilde{m}_{22} + \tilde{m}_{23}) + D_z \cot\alpha\right]\frac{W_j}{L_j}} \tag{8}$$

where $\tilde{m}_{ij}$ is the element of the inverse matrix of **M**.

It is seen that the variables except x, y, W and L in the above formula are constants for a single camera. The vehicle width W is often small enough to be neglected relative to the other terms of the numerator. It is a reasonable assumption that the change in the ratio of the width to the length of a vehicle class is very small. For each class of vehicle, the correction factor R can be approximately expressed as a polynomial in the following form.

$$R_j = \frac{A_1 x_1^j + B_1 y_1^j + C_1}{A_2 x_1^j + B_2 x_1^j + C_2} \tag{9}$$

The parameters (A_1, A_2, B_1, B_2, C_1, C_2) can be determined by performing a regression on the coordinates data in the dataset Ve-YJ. For the j-th vehicle, the final coordinates used to calculate the transverse position are (x_{center}^j, y_1^j), in which x_{center}^j is obtained by

$$x_{center}^j = \frac{(x_1^j + x_3^j)}{2} - \frac{R_j(x_3^j - x_1^j)}{2} \tag{10}$$

With the correction factor R, it becomes possible to only detect vehicles without the need to detect their front or rear, greatly saving the calculation time. A test of a video of the scene in Fig.9 showed that the calculation time can be reduced by about 40% from a 0.05 sec/frame to a 0.029 sec/frame.

The relative errors of the factor R were investigated for all vehicles in the dataset Ve-YJ, and shown in Table 3 according to the camera number and vehicle type.

Table 3. The relative errors of R for different vehicles and FOVs

Types	*Car& Front*	*Car& Rear*	*Bus& Front*	*Bus& Rear*	*Truck& Front*	*Truck& Rear*	*No. of FOVs*
Error_{max}	12.59%	12.56%	7.89%	11.51%	11.49%	12.93%	Camera 1

Error$_{average}$	2.53%	2.59%	2.08%	2.10%	2.82%	2.79%	
Pr (Error>10%)	0.28%	0.50%	0.00%	0.76%	1.10%	1.64%	
Error$_{max}$	14.11%	15..17%	10.44%	9.88%	9.53%	12.06%	Camera 2
Error$_{average}$	2.72%	2.73%	2.02%	2.08%	3.15%	3.22%	
Pr (Error>10%)	0.84%	0.68%	0.80%	0.00%	0.00%	**3.03%**	
Error$_{max}$	13.44%	13.99%	10.94%	**17.00%**	10.34%	11.24%	Camera 3
Error$_{average}$	2.09%	**3.86%**	2.34%	2.20%	3.36%	3.25%	
Pr (Error>10%)	0.10%	1.21%	0.39%	0.87%	0.71%	1.20%	
Error$_{max}$	10.76%	13.67%	8.51%	10.66%	7.13%	10.19%	Camera 4
Error$_{average}$	1.65%	2.04%	2.59%	2.32%	3.02%	3.00%	
Pr (Error>10%)	0.08%	0.31%	0.00%	0.60%	0.00%	1.33%	

Note:. *'Vehicle'&Front* represents the oncoming vehicle with a visible front, and *'Vehicle'&Rear* the departing vehicle with a visible rear. *Pr (Error>10%)* indicates the probability that the relative error is greater than 10%.

It can be seen from the results that the maximum relative error of the correction factor is 17%, the average relative error is about 3%, and the probability of the relative error exceeding 10% is also very low (<1% in most cases).

7 Coordinate transformation from image to bridge coordinates

Identification of the external and internal camera parameters are required for transformation from image coordinates to bridge coordinates. External camera parameters refer to the transformation matrix between the camera image coordinates and bridge deck coordinates. In general, the transformation matrix is estimated using the coordinate pairs of the image coordinates and corresponding bridge deck coordinates of reference points arranged on the bridge deck, which has been introduced in detail in [18]. In this study, a field test was conducted on the Yongjiang Bridge with a total of 49 marked PVC boards arranged along the bridge (Fig.12). The precise coordinates of the reference points on the bridge deck were all measured by a total station, while their image coordinates were manually read from the images.

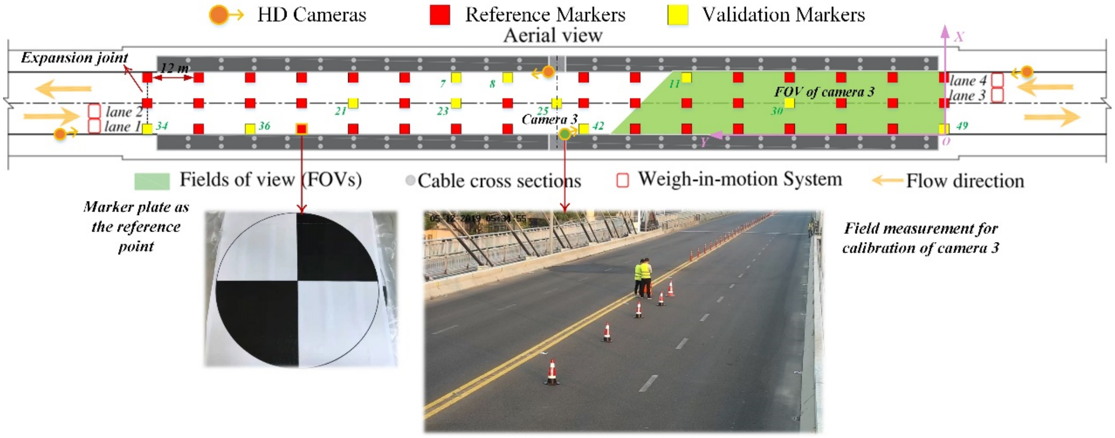


Figure 12. Field test for the calculation of external parameters

The camera internal parameters were estimated using a checkerboard and Zhang method [23] to correct the lens distortion (Fig.13).

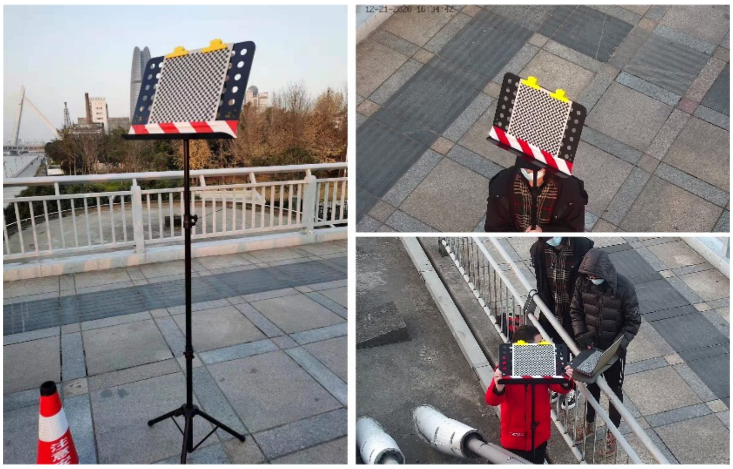


Figure 13. Field test for the calibration of internal parameters

In order to verify the accuracy of the coordinates' transformation matrices identified from the field test, three reference points in each FOV were excluded from the calculation of transformation matrix (eleven yellow markers in Fig.12), and the errors were calculated as shown in Table 4. It showed the errors of the eleven yellow reference points and the average and maximum errors of all reference points, where X_{error} indicated the errors in the transverse bridge direction and Y_{error} in the longitudinal bridge direction.

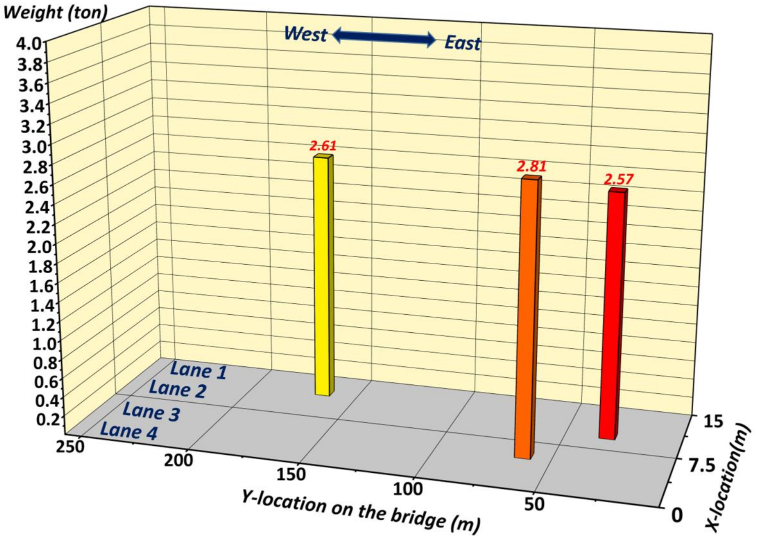
Table 4. Transformation errors of reference points in the field test (unit: m)

No. of reference points	FOVs	$X_{\text{yellow error}}$	$Y_{\text{yellow error}}$	$X_{\text{average error}}$	$Y_{\text{average error}}$	$X_{\text{maximum error}}$	$Y_{\text{maximum error}}$
8	Camera 1	0.199	0.811	0.047	0.225	0.120	0.458
23		0.062	0.509				
36		0.000	0.128				
7	Camera 2	0.015	0.166	0.055	0.360	0.183	0.943
21		0.018	0.103				
34		0.144	1.153				
11	Camera 3	0.332	0.108	**0.059**	**0.473**	0.206	**1.350**
30		0.032	0.243				
49		0.006	0.043				
11	Camera 4	0.059	0.425	0.051	0.266	**0.216**	0.657
25		0.077	0.288				
42		0.196	0.610				

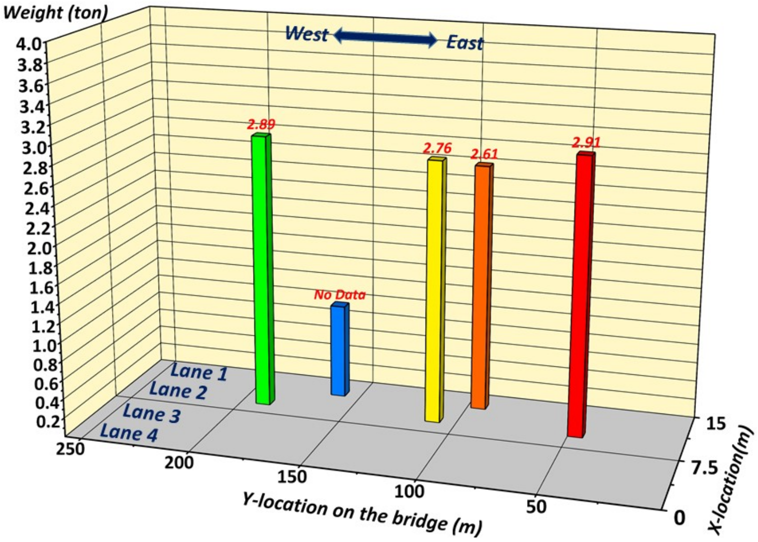
From the results, the average value of the coordinate transformation error of the reference points was about 0.05m in the transverse direction and less than 0.5m in the longitudinal direction. As for the maximum error, it was 0.216m in transverse direction while 1.35m in longitudinal direction.

7.2 Full-bridge traffic load distribution estimation result

Twenty four hours of videos and WIMs text data collected on November 16, 2020 and four-minute continuous segments from 4:49 p.m to 4:53 p.m were selected for a demonstration of the proposed long-term FBTL monitoring system. Figure 14 shows the visualisation of analysis results for two calculation frames (more results can be seen in the attached videos).



(a) The 2302*th* calculation frame



(b) The 4420*th* calculation frame

Figure.14. The full-bridge traffic load distribution on Yongjiang Bridge

In Fig.14, the bridge deck of Yongjiang Bridge is represented by the X-Y plane, vehicle centroid by the centre of the bottom surface of the square column, and the load value (vehicle weight) by column height. It is worth noting that, in most cases, a complete traffic load distribution on the bridge was estimated as shown in Fig.14(a). However, in a few cases, the WIM system may have missed detection on the vehicle weights, resulting in the absence of load value as shown in Fig.14(b). The vehicle indicated by the blue bar had no weight attached while all other vehicles in the same lane did. In the four minutes, a total of 36 vehicles passed through lanes 1, 2 and 3, while 4 vehicles were missed by the WIM system. The WIMs in lane 4 failed during the test period and were not included in the statistics.

To summarise from the cases discussed above, even if the current WIM technology has been well developed, there may still be detection failures or missed detections because of the combined influence of installation quality, operating environment and traffic conditions. A special design in the information fusion has been made, so that the proposed system can accurately match the detected vehicle to the corresponding weight information, but also automatically identify and label the missing WIM data based on the video analysis results and time clues.

8 Conclusion

This study proposed a long-term full-bridge traffic load monitoring framework and demonstrated its first application on a long-span cable-stayed bridge. Three improved methods were proposed for accurate vehicle detection, robust vehicle tracking, and fast vehicle positioning error correction as summarised below.

- Novel methods for vehicle detection and tracking based on deep CNNs were developed on the image dataset (Ve-YJ) dedicated to bridge traffic scenes, greatly improving the ability of the existing FBTL framework to cope with challenging long-term monitoring tasks.
- The proposed correction method of vehicle transverse location achieved a faster computation speed and is suited to cases where a large number of targets need to be processed.
- The relevant algorithms of the proposed FBTL monitoring system have been demonstrated on a cable-stayed bridge, and the system is also applicable to any other forms of bridges (suspension bridges, variable width bridges and even curved bridges).

Combined with the monitoring of bridge structural response, the proposed system is expected to be used as a tool in some important issues, such as load analysis, model updating and damage identification. With the rapid development of deep learning and sensing technologies, the proposed framework could be further improved in the future.

Acknowledgement

This work is supported by the National key R&D Program of China (2017YFF0205605); the National Nature Science Foundation of China (Grant No. 5187849); Shanghai Urban Construction Design Research Institute Project ‘Bridge Safe Operation Big Data Acquisition Technology and Structure Monitoring System Research’; the Ministry of Transport Construction Science and Technology Project ‘Medium-Small Span Bridge Structure Network Level Safety Monitoring and Evaluation’; and the CSC (China Scholarship Council) Scholarship (202006260246).

References

[1] OBrien, E. J., Schmidt, F., Hajializadeh, D., Zhou, X. Y., Enright, B., Caprani, C. C., ... & Sheils, E. (2015). A review of probabilistic methods of assessment of load effects in bridges. Structural safety, 53, 44-56.

[2] Deng, Y., Li, A., Chen, S., & Feng, D. (2018). Serviceability assessment for long‐span suspension bridge based on deflection measurements. Structural Control and Health Monitoring, 25(11), e2254.

[3] Deng, L., & Cai, C. S. (2010). Bridge model updating using response surface method and genetic algorithm. Journal of Bridge Engineering, 15(5), 553-564..

[4] Mao, Q., Mazzotti, M., DeVitis, J., Braley, J., Young, C., Sjoblom, K., ... & Bartoli, I. (2019). Structural condition assessment of a bridge pier: A case study using experimental modal analysis and finite element model updating. Structural Control and Health Monitoring, 26(1), e2273.

[5] Nie, Z., Lin, J., Li, J., Hao, H., & Ma, H. (2020). Bridge condition monitoring under moving loads using two sensor measurements. Structural Health Monitoring, 19(3), 917-937.

[6] Feng, K., Gonzɑ́lez, A., & Casero, M. (2021). A kNN algorithm for locating and quantifying stiffness loss in a bridge from the forced vibration due to a truck crossing at low speed. Mechanical Systems and Signal Processing, 154, 107599.

[7] Yu, Y., Cai, C. S., & Deng, L. (2016). State-of-the-art review on bridge weigh-in-motion technology. Advances in Structural Engineering, 19(9), 1514-1530.

[8] Deng, L., & Cai, C. S. (2011). Identification of dynamic vehicular axle loads: Demonstration by a field study. Journal of Vibration and Control, 17(2), 183-195.

[9] Yu, Y., Cai, C. S., & Deng, L. (2017). Vehicle axle identification using wavelet analysis of bridge global responses. Journal of Vibration and Control, 23(17), 2830-2840.

[10] OBrien, E. J., Lipari, A., & Caprani, C. C. (2015). Micro-simulation of single-lane traffic to identify critical loading conditions for long-span bridges. Engineering Structures, 94, 137-148.

[11] Zhou, J., Ruan, X., Shi, X., & Caprani, C. C. (2019). An efficient approach for traffic load modelling of long span bridges. Structure and Infrastructure Engineering, 15(5), 569-581.

[12] Ge, L, Dan, D, Z, Liu, &X, Ruan. (2021). Intelligent Simulation Method of Bridge Traffic Flow Load Combining Machine Vision and Weigh-in-motion Monitoring. (under review)

[13] Hester, D., Koo, K., Xu, Y., Brownjohn, J., & Bocian, M. (2019). Boundary condition focused finite element model updating for bridges. Engineering Structures, 198, 109514.

[14] Conte, J. P., He, X., Moaveni, B., Masri, S. F., Caffrey, J. P., Wahbeh, M., ... & Elgamal, A. (2008). Dynamic testing of Alfred Zampa memorial bridge. Journal of structural engineering, 134(6), 1006-1015.

[15] Ren, W. X., Lin, Y. Q., & Peng, X. L. (2007). Field load tests and numerical

analysis of Qingzhou cable-stayed bridge. Journal of Bridge Engineering, 12(2), 261-270.

[16] Brown, R., & Wicks, A. (2016). Vehicle tracking for bridge load dynamics using vision techniques. In Structural Health Monitoring, Damage Detection & Mechatronics, Volume 7 (pp. 83-90). Springer, Cham.

[17] Chen, Z., Li, H., Bao, Y., Li, N., & Jin, Y. (2016). Identification of spatio‐temporal distribution of vehicle loads on long‐span bridges using computer vision technology. Structural Control and Health Monitoring, 23(3), 517-534.

[18] Dan, D., Ge, L., & Yan, X. (2019). Identification of moving loads based on the information fusion of weigh-in-motion system and multiple camera machine vision. Measurement, 144, 155-166.

[19] Zhou, Y., Pei, Y., Li, Z., Fang, L., Zhao, Y., & Yi, W. (2020). Vehicle weight identification system for spatiotemporal load distribution on bridges based on non-contact machine vision technology and deep learning algorithms. Measurement, 159, 107801.

[20] Ge, L., Dan, D., & Li, H. (2020). An accurate and robust monitoring method of full‐bridge traffic load distribution based on YOLO‐v3 machine vision. Structural Control and Health Monitoring, 27(12), e2636.

[21] Ge, L., Dan, D., Yan, X., & Zhang, K. (2020). Real time monitoring and evaluation of overturning risk of single-column-pier box-girder bridges based on identification of spatial distribution of moving loads. Engineering Structures, 210, 110383.

[22] Guo, H., & Crossley, P. (2016). Design of a time synchronization system based on GPS and IEEE 1588 for transmission substations. IEEE Transactions on power delivery, 32(4), 2091-2100.

[23] Liu, X., Liu, W., Mei, T., & Ma, H. (2016, October). A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In European conference on computer vision (pp. 869-884). Springer, Cham.

[24] Zhang, Z. (2000). A flexible new technique for camera calibration. IEEE Transactions on pattern analysis and machine intelligence, 22(11), 1330-1334.

[25] Zhang, B., Zhou, L., & Zhang, J. (2019). A methodology for obtaining spatiotemporal information of the vehicles on bridges based on computer vision. Computer‐Aided Civil and Infrastructure Engineering, 34(6), 471-487.

[26] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[27] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[28] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916.

[29] Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition

workshops (pp. 390-391).

[30] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020, April). Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 07, pp. 12993-13000).

[31] Lowe, D. G. (1999, September). Object recognition from local scale-invariant features. In Proceedings of the seventh IEEE international conference on computer vision (Vol. 2, pp. 1150-1157). Ieee.

[32] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). Ieee.

[33] Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., & Tian, Q. (2016, October). Mars: A video benchmark for large-scale person re-identification. In European Conference on Computer Vision (pp. 868-884). Springer, Cham.

[34] Wojke, N., & Bewley, A. (2018, March). Deep cosine metric learning for person re-identification. In 2018 IEEE winter conference on applications of computer vision (WACV) (pp. 748-756). IEEE.