sensors

MDPI

Article

Identification and Tracking of Vehicles between Multiple Cameras on Bridges Using a YOLOv4 and OSNet-Based Method

Tao Jin, Xiaowei Ye *, Zhexun Li and Zhaoyu Huo

Department of Civil Engineering, Zhejiang University, Hangzhou 310058, China; cetaojin@zju.edu.cn (T.J.)
* Correspondence: cexwye@zju.edu.cn

Abstract: The estimation of vehicle loads is a rising research hotspot in bridge structure health monitoring (SHM). Traditional methods, such as the bridge weight-in-motion system (BWIM), are widely used but they fail to record the locations of vehicles on the bridges. Computer vision-based approaches are promising ways for vehicle tracking on bridges. Nevertheless, keeping track of vehicles from the video frames of multiple cameras without an overlapped visual field poses a challenge for the tracking of vehicles across the whole bridge. In this study, a method that was You Only Look Once v4 (YOLOv4)- and Omni-Scale Net (OSNet)-based was proposed to realize vehicle detecting and tracking across multiple cameras. A modified IoU-based tracking method was proposed to track a vehicle in adjacent video frames from the same camera, which takes both the appearance of vehicles and overlapping rates between the vehicle bounding boxes into consideration. The Hungary algorithm was adopted to match vehicle photos in various videos. Moreover, a dataset with 25,080 images of 1727 vehicles for vehicle identification was established to train and evaluate four models. Field validation experiments based on videos from three surveillance cameras were conducted to validate the proposed method. Experimental results show that the proposed method has an accuracy of 97.7% in terms of vehicle tracking in the visual field of a single camera and over 92.5% in tracking across multiple cameras, which can contribute to the acquisition of the temporal–spatial distribution of vehicle loads on the whole bridge.

Keywords: structural health monitoring; deep learning; temporal–spatial distribution; vehicle loads; vehicle identification

Citation: Jin, T.; Ye, X.; Li, Z.; Huo, Z. Identification and Tracking of Vehicles between Multiple Cameras on Bridges Using a YOLOv4 and OSNet-Based Method. *Sensors* **2023**, *23*, 5510. https://doi.org/10.3390/s23125510

Academic Editors: Giuseppe Lacidogna, Sanichiro Yoshida, Guang-Liang Feng, Jie Xu, Alessandro Grazzini and Gianfranco Piana

Received: 11 May 2023
Revised: 4 June 2023
Accepted: 6 June 2023
Published: 12 June 2023

1. Introduction

Bridge structures are critical components of transportation infrastructures that contribute to the smoothness of traffic flow. However, as time goes by, the safety of in-service bridges is challenged by the effects of multiple factors, especially vehicle loads [1–4]. On one hand, the load standards for the design of bridges were determined decades ago yet there are more vehicles and heavier vehicles on the bridges nowadays. On the other hand, eccentric loading of bridges induced by the unilateral passage of heavy vehicles, such as trucks and flat cars, will also lead to damage to bridge components [5,6]. Therefore, the acquisition of vehicle loads on the bridges is vital to decisions regarding maintenance for structural safety.

Thanks to the development of sensing techniques and data processing algorithms, SHM-based methods have been proposed to detect vehicle loads on bridges. Among the many kinds of techniques for sensing vehicle loads, the bridge weight-in-motion system (BWIMs) proposed by Moses [7] is preferred for practical application by those in the industrial community [8,9]. The BWIMs applies sensors to capture bridge stress and strain, and analyzes its dynamic strain responses to restore vehicle information, including vehicle load, speed, number of axles, etc. [10]. Its high precision and broad applicability has attracted many scholars. Wu et al. [11] developed an encoder–decoder structure called BwimNet to identify the properties of moving vehicles. The model is multi-target and

can detect axle number, speed, weight, and wheelbases simultaneously. After decades of development, the BWIMs present excellent performance and has been installed on many major bridges. However, the BWIMs are fixed in a certain section and can detect the vehicle loads only when vehicles are passing over the sensors embedded under the bridge deck [12]. Thus, the temporal–spatial distribution of vehicle loads on bridges is not available when using the BWIMs.

In recent years, many researchers have adopted vision-based methods to locate vehicles due to the rapid development of computer vision techniques [13–18]. Chen et al. [19] proposed a Gaussian Mixture Model (GMM) and a shadow removal method-based approach to detect and track vehicles through CCTV devices. Chen et al. [20] established a real-time vehicle detection and counting method based on single shot detection (SSD) and reached an accuracy of 99.3%. The vehicles were classed into six groups, including cars, taxis, vans, trucks, motorbikes, and buses. Harikrishnan et al. [21] put forward a bounding box algorithm to locate vehicles, which was estimated with two-dimensional binary histogram projection profile (2D-BHPP) algorithm. Zhang et al. [22] developed a vehicle detection algorithm based on the Faster region-based convolutional neural network (Faster R-CNN), and the Zeiler and Fergus model (ZF). The method was applied to automatically detect vehicle types, number of axles, and length for the temporal–spatial information of vehicles on bridges. The computer vision-based vehicle detection approaches could obtain the temporal–spatial distribution of vehicle loads in the field of a single camera which covers a limited portion of the whole bridge. When the visual fields of multiple cameras are not continuous, the shapes of the same vehicle in video frames of different cameras will be quite different, which challenges the detection of the vehicles along the whole bridge, as illustrated in Figure 1.

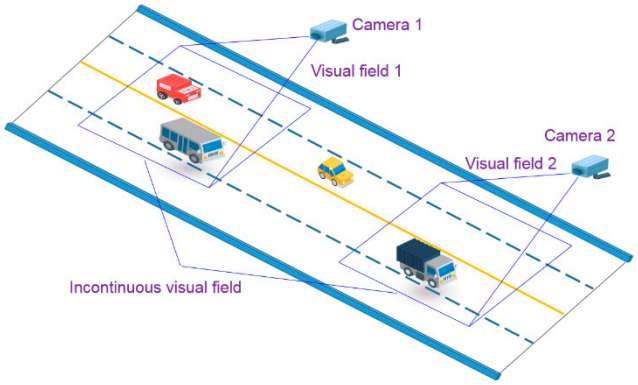


Figure 1. Challenges of computer vision-based vehicle detection.

To overcome the existing problem, Chen et al. [23] applied feature and area-based approaches to re-identify vehicles between multiple cameras. Edge detection was applied to extract the features of vehicles, and a template matching algorithm was used to track vehicles. The temporal–spatial distribution model of traffic loads on Hangzhou Bay Bridge was obtained. Yet, the template matching algorithm is sensitive to the quality of the captured video images. Dan et al. [24] used Kalman Filter to track a vehicle and calculated the time it should appear in the visual field of the next camera. The best matched one would be considered as the same vehicle, but it could make false predictions when the distances between the adjacent vehicles are small.

The vehicle re-identification between multiple cameras is a hot field in computer vision [25–27]. The number of re-identification studies has grown in number, aiming to solving the challenge of matching objects across different cameras when the primary hallmark, such as the face or plate number, is unrecognized. Many neural network models that focus on the re-identification issue have been put forward [28–30], and due to the distinctiveness of vehicles, additional information has been applied for precise re-identification. The Siamese-CNN + Path-LSTM model proposed by Shen et al. [31] takes the vehicle

path into account. These studies mainly concentrate on modifying models and methods to improve performance in the existing dataset, such as VeRi-776 [32], CompCars [33], and VERI-Wild [34].

Inspired by the re-identification method, a YOLOv4 and OSNet-based method for identification of the temporal–spatial distribution of vehicle loads on bridges was proposed in this study. It includes a YOLOv4-based vehicle detection module and an OSNet-based feature extraction and re-identification module. A dataset with 25,080 images related to 1727 vehicles was established to train the OSNet and a field validation experiment was conducted to test the proposed method for evaluation of robustness and reliability.

2. Framework of the Proposed Method

The proposed vehicle detection, tracking, and re-identification method based on the YOLOv4 neural network and the OSNet is shown in Figure 2. The proposed method is mainly composed of two modules, one is for vehicle detection from the video images and the other is for re-identification of the same vehicle from the video images captured by different cameras.

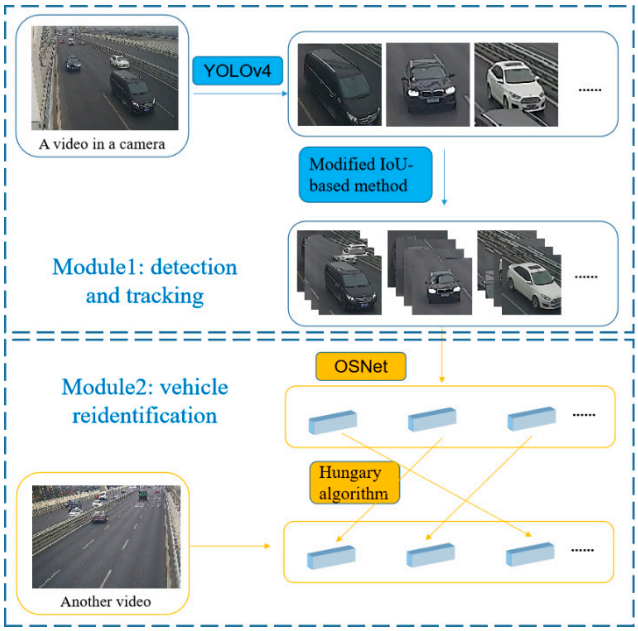


Figure 2. Framework of the proposed method.

In the first module, YOLOv4 is applied to detect and locate vehicles, while the Kalman Filter and a modified IoU-based tracking method are used to track the same vehicle between adjacent frames captured by the same camera. Every vehicle with corresponding information regarding location and time will be stored and post-processing is adopted to suppress the interference of incorrect detection. In addition, a novel algorithm for position correction was proposed to clear the lanes of vehicles. The second module is a vehicle re-identification module, which adopts OSNet, to extract features of vehicle images. After that, a re-ranking method was introduced to amend the Euclidean distance between image features, and the Hungary algorithm proposed by Munkres [35] was adopted to match the same vehicle from images captured by multiple cameras without an overlapped visual field.

3. Vehicle Detection and Tracking with a Single Camera

Vehicle detection and tracking based on a single camera is the foundation of the proposed method. The flowchart for processing the video frames from the same camera for vehicle detection and tracking is shown in Figure 3. Vehicle detection is conducted by the YOLOv4 and the tracking of vehicles is realized with a modified IoU-based method.

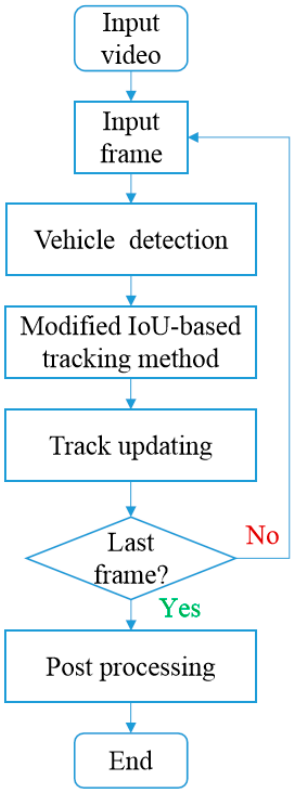


Figure 3. Flowchart of vehicle detection and tracking.

3.1. Architecture of YOLOv4

The YOLOv4 proposed by Bochkovskiy et al. [36] has been adopted by many researchers for satisfactory performance [37,38] and has been utilized in this study. Shown in Figure 4, the architecture of YOLOv4 contains three parts, CSPDarknet53 as the backbone, SPP + PAN as the neck, and YOLOv3 as the head. In the SPP, there are three pooling channels with different kernel sizes, which are 5×5, 9×9 and 13×13.

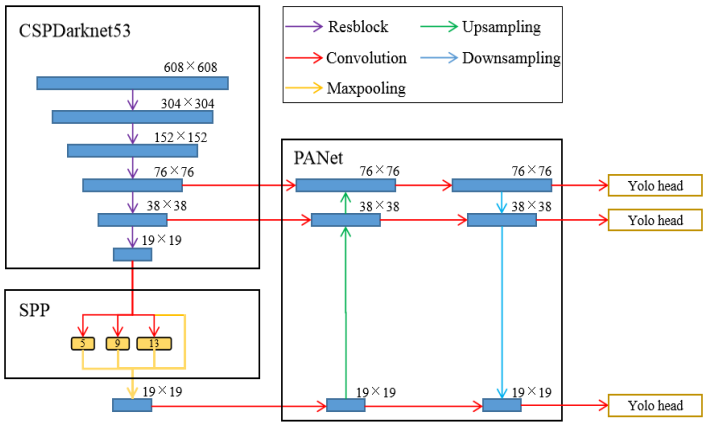


Figure 4. The architecture of the YOLOv4.

The CSPDarknet53 adopted a cross-stage feature fusion strategy named Cross Stage Partial (CSP) proposed by Wang et al. [39]. The feature map is divided into two parts, one part goes through convolution layers and the other passes through a shortcut and concatenates with the former. Thus, it reduces computations without degradation of detection ability. CSPDarknet53 consists of Darknet53, the backbone of YoLOv3, and CSP to achieve fast and precise detection. The Spatial Pyramid Pooling (SPP) method proposed by He et al. [40] and the Pixel Aggregation Network (PAN) proposed by Wang et al. [39]

are combined to fuse features at different scales. They allow YOLOv4 to transmit features from various layers and benefits the feature extraction. The prediction module of YOLOv4 applies three scales to detect objects with different sizes, and outputs their positions, categories, and confidences.

Moreover, a batch of methods are tested in YOLOv4 to achieve a higher level of detection. These methods cover activations, bounding box regression loss, data augmentation, and so on. Among them, Distance-IoU (DIoU) contributes a lot to reducing the possibility of low recall, and has a higher potential in vehicle detection. The previous non-maximum suppression (NMS) algorithm only adopts intersection over union (IoU) to remove redundant bounding boxes and retain the one that is most possible. In response to this, DIoU takes the distance of box centers into consideration along with IoU, and reduces the mistaken elimination, as shown in Figure 5.

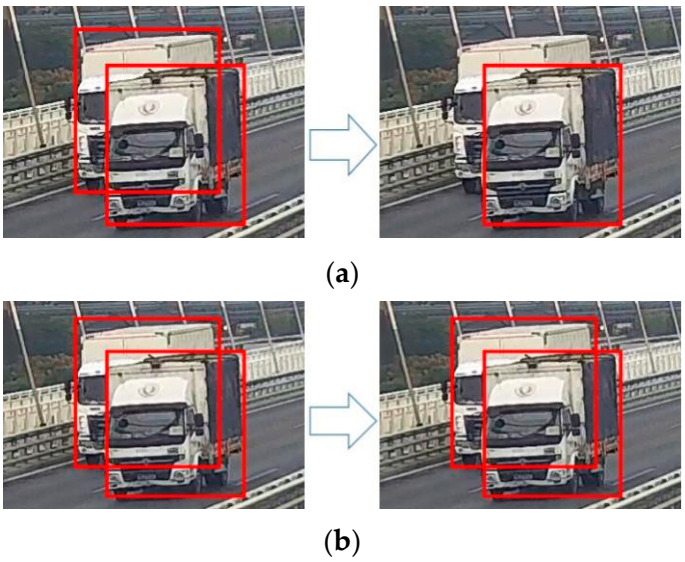


Figure 5. Difference between IoU-NMS and DIoU-NMS algorithm: (**a**) IoU-NMS algorithm; (**b**) DIoU-NMS algorithm.

The original resolution of the surveillance videos is 1080 × 1920, which does not suit the requirement of YOLOv4. In order to achieve a balance between speed and accuracy, the frames were resized into 512 × 896 resolutions, which maintains the original aspect ratio as much as possible.

3.2. Modified IoU-Based Tracking Method

Tracking vehicles between adjacent frames meets the challenges of multiple vehicles, missing detection, and false-positive detection. In order to overcome these challenges, Chen et al. [20] applied a bounding box distance between the box center of consecutive sequence of frames to implement vehicle tracking. However, the neglect of vehicle movement limits its recognition capability, and for a bounding box, there is only one point used to track, which will be often disturbed by bounding boxes of different sizes. Zhou et al. [16] adopted the Kalman filter to predict vehicle positions in the current frame depending on the former tracks, and applied the predicted box to track vehicles. Previous research mainly focused on spatial information of vehicles while appearance features of the vehicles in the bounding boxes can make a contribution to vehicle tracking as well [41]. Therefore, a modified IoU-based tracking method which takes appearance into consideration was proposed, as shown in Figure 6.

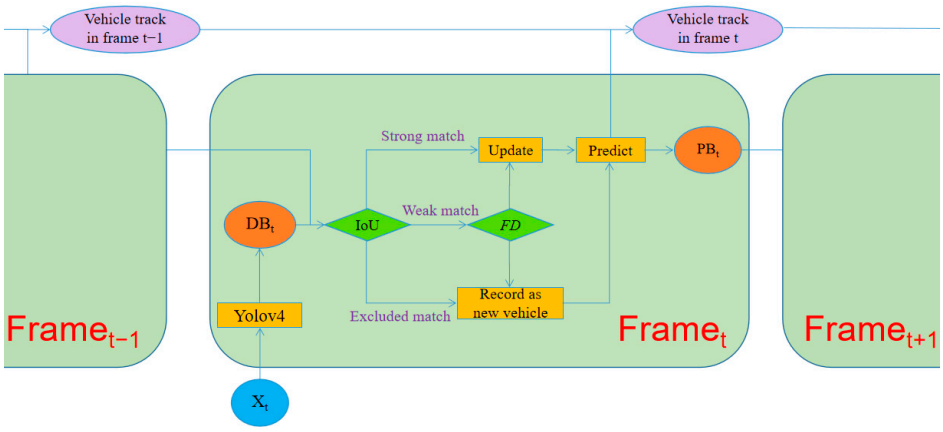


Figure 6. Modified IoU-based tracking method.

During the processing of vehicle detection via YOLOv4, a batch of rectangular object proposals with corresponding scores and categories were obtained, and these bounding boxes will be compared with the bounding boxes predicted in the last frame, as Figure 7a shows. IoU is used to estimate their overlap ratio and calculated as Figure 7b. Each bounding box of the current frame will obtain a list of IoU values which represents its overlapping ratio with the last determined bounding boxes, and the maximum will be chosen as the most likely one for multiple vehicles. Matched results are classified into three classes, the strong match, the weak match, and the excluded match. A strong match whose IoU value is over 0.4 means two bounding boxes are likely to represent the same vehicle. Afterwards, the corrected detection result based on prediction will be calculated and appended to the track of the detected vehicle. Moreover, an additional bounding box will be predicted for tracking in the next frame. An excluded match means the detecting bounding box does not overlap with any of the predicted boxes and suggests a new vehicle might appear in this frame. Then, a new series number will be generated to refer to this vehicle, and the exterior information extracted by the OSNet will be bound to this vehicle. It should be explained that every bounding box in the first frame is regarded as an excluded match.

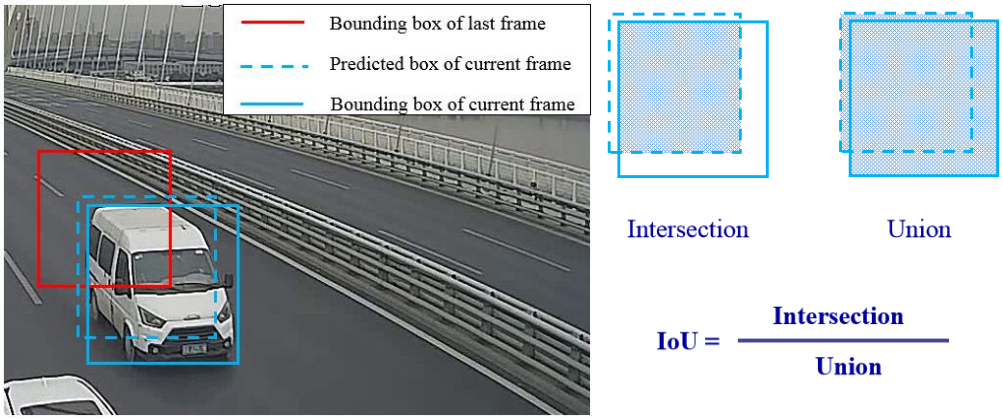


Figure 7. Vehicle detection with the modified IoU-based tracking method: (**a**) the abridged general view of bounding boxes; (**b**) the formula for calculating the IoU.

The main difficulty comes from the weak match whose IoU value is between 0 and 0.4. External information is adopted for accurate judgment. The current vehicle image will be input into the OSNet and the characteristics of the appearance of vehicles will be obtained. Then, Euclidean distance between the characteristics of the appearance of

two vehicle images could be calculated. Due to the similar direction of the visual field in a single camera, the images of the same vehicle usually share a similar appearance, and their Euclidean distance would be significantly smaller than that of different vehicles. With this approach, the classification of weak matches is obtained. The tracking method is a recurrent procedure which will keep working until all the video frames are processed as shown in Figure 3.

3.3. Kalman Filter

The Kalman filter is a recursive method that estimates the state of the target object combined with prediction and detection results [42]. An 8-dimensional state space z for prediction is established as follows.

$$x = \left[m, n, a, h, m', n', a', h'\right] \tag{1}$$

where m and n refer to the bounding box center position, a stands for the aspect ratio, and h is the height of the bounding box. The next couple of terms represent their derivatives, respectively. Forecasts to draw a predicted box are performed by using the following equation:

$$x_i{}' = Ax_{i-1} \tag{2}$$

$$P_i' = AP_{i-1}A^T + Q \tag{3}$$

The matrix A relates the state at the previous time step to the current step, and the state covariance matrix of the last time step P_{i-1} is used to calculate that in the next step P_i. Q stands for indeterminacy of state.

Updating operation works after prediction:

$$K_i = P_i'H^T\left(HP_i'H^T + R\right)^{-1} \tag{4}$$

$$x_i = x_i{}' + K\left(z_i - Hx_i{}'\right) \tag{5}$$

$$P_i = (I - K_iH)P_iH' \tag{6}$$

The first step of updating is to calculate the Kalman gain K, and the measurement matrix H relates measurement to state. R stands for noises of devices. The second step is to gain the revised state estimate based on the measurement z and predicted state $x_i{}'$. Finally, a posteriori error covariance estimate is obtained.

3.4. Adaptive Lane Division Method

There are a lot of studies on conversion from the camera coordinate system to the world coordinate system [17,43]. However, the center or the bottom of the bounding box cannot represent the position of the target vehicle precisely (Figure 8), and may lead to the error distinction.

Figure 8. Mismatch between the bounding box and lane.

In this section, an alternative approach for lane division was proposed. A batch of bounding boxes on pilot run of YOLOv4 with no tracking operation can be obtained, and their midpoints of rectangular bottom lines can be noted, as shown in Figure 9. The dots converge into six lines, each representing their corresponding lane. Red lines shown in Figure 8 are the ground-truth axes of lanes and they do not match dots completely. The yellow lines are the centers of the dots shown in Figure 9. Therefore, we applied a two-step method, firstly (i) classify dots depending on the distance from the points to the base-lines and then (ii) every group is used to fit a new base-line for the least variance.

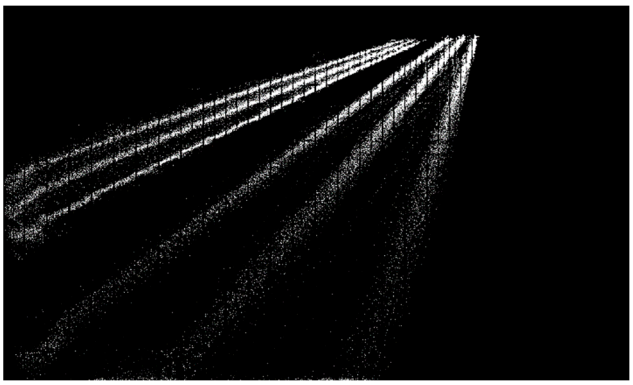


Figure 9. The tracks of the detected vehicles.

The method will be repeated until the result is almost unchanged. Compared with the initial lines in red, yellow lines are more suitable for lane division, especially for the one far away from the camera.

4. OSNet-Based Vehicle Re-Identification

The OSNet is a convolutional neural network (CNN) focusing on re-identification issues. The OSNet can capture different spatial scales, and integrate these scales as output.

4.1. The Bottleneck of the OSNet

Figure 10 shows the bottleneck of the OSNet. It is evident that the bottleneck contains four paths with different numbers of convolution layers, thus various scale features are obtained. The numbers are the amount of lite convolutions in different paths. Aggregation gate (AG) refers to the unified aggregation gate, which controls weights assigned to different scales. The aggregation gate is novel to others in terms of its ability to learn, which requires less human intervention. The lite convolution separates a standard convolution layer into a pointwise layer and a depthwise layer. In the case that the result of calculation is slightly changed, this operation significantly reduces the amount of calculation. The OSNet network will automatically select different outputs according to the model state. During the training stage, features (when triple loss is adopted) and categories will be output, and during the test stage, only features will be output.

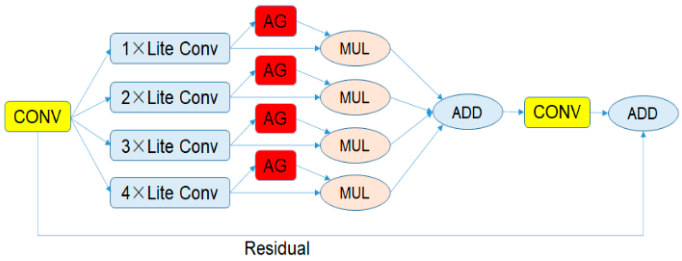


Figure 10. The bottleneck of the OSNet.

4.2. Establishment of Dataset

A vehicle re-identification dataset is established to train the OSNet. Pictures in the dataset were captured as explained in Section 2 with the exception of the classification of weak matches. In the process of dataset establishment, all the weak matches were considered as excluded matches, and two members of this study checked them manually. After making sure the vehicle images of a single camera were properly classified, vehicle images from different camera frames were labeled. Thus, every vehicle in the established dataset was captured by at least two cameras and all the pictures of it share the same label. Finally, 25,080 images of 1727 vehicles were collected and labeled, for each of which there was a camera marking. Figure 11 shows examples of the established dataset.

Figure 11. Samples of the established vehicle dataset.

The dataset is divided into two subsets for training and testing, respectively. The training set contains 1727 vehicles with 25,080 images and the testing set has 200 vehicles with 2891 images. Additionally, open-sourced datasets VeRi-776 and VERI-Wild were utilized for a comparison study.

4.3. Data Processing

For this study, the size of the input images was set as 256 × 256 resolutions, which fits the shape of vehicle images better. In order to improve the OSNet's robustness, the images were resized into 288 × 288 resolutions and randomly cropped to 256 × 256 resolutions. Horizontal flip at a possibility of 0.5 was adopted also for augment processing in this study.

4.4. Evaluating Indices

In this study, the Rank-1 and the mean average precision (mAP) index were adopted as the evaluation of vehicle identification performance. Three other re-identification models including Res50-dim [44], PCB [45] and HA-CNN [46] were applied as contrasts. In the training process, a query picture was input, and the neural network searched the picture belonging to the same category from the gallery images and ranked them according to their probability. Rank-1 represents the rate that the picture with the highest probability and query one are indeed the same vehicle. The mAP was used to denote the precision of the neural network as follows.

$$mAP = \frac{\sum_{k=1}^{C} AP_k}{C} \tag{7}$$

$$AP = \frac{\sum_{i=1}^{n} Precision_i}{n} \tag{8}$$

$$Precision_i = \frac{i}{Position_i} \tag{9}$$

Every registered vehicle has a couple of pictures with the same label, and in Equation (8), the n stands for the number of this category, and $Precision_i$ is the precision of the ith picture of the registered vehicle.

4.5. Hyper-Parameter

Adam proposed by Kingma and Ba [47] is adopted as the optimizer in this paper. The cross entropy loss and triplet loss were adopted for weight updating. The cross entropy loss is calculated as follows

$$H(p,q) = -\sum(p(x)\log q(x)) \tag{10}$$

where $H(p, q)$ is the cross entropy loss applied to updating the neural network weights along with triplet loss. $p(x)$ represents the label of input vehicle image, and $q(x)$ is the output possibility of if the input vehicle is the same one as the label.

Triplet loss was proposed for face re-identification by Schroff et al. [48]. It aims to shorten the distance of features from the same category and enlarge the distance from various types. It is defined by

$$L = \max(d(a,p) - d(a,n) + margin, 0) \tag{11}$$

where a means anchor, p stands for positive sample, and n stands for negative sample. $margin$ is a constant usually defaulted as 0.3.

The hyperparameter is summarized in Table 1 and was utilized to train four different models.

Table 1. Hyperparameter for training models.

Size of Input	Max Epoch	Batch Size	Optimizer	Initial Learning Rate	Loss Function
256 × 256	60	32	Adam	0.0003	Triplet Loss + Cross Entropy Loss

4.6. Training Results

The OSNet was trained in a workstation, and its hardware and software are listed in Table 2. Torchreid is a software library based on Pytorch, and it allows convenient training and evaluation of re-identification models.

Table 2. Hardware and software for training.

Item	Version
Hardware	CPU: 2 × Intel(R) Xeon(R) Silver4215R CPU @ 3.20 GHz GPU: NVIDIA RTX 3090/GDDR5X 24 GB RAM: 64 GB
Software	Windows 10 Version 1909 Pytorch 1.7.1 + cu110 Python 3.8.5 Opencv 4.4.0 Torchreid 1.3.3

Figure 12 shows the evaluation indicators of diverse models, and obviously, the OSNet surpasses other methods by a clear margin.

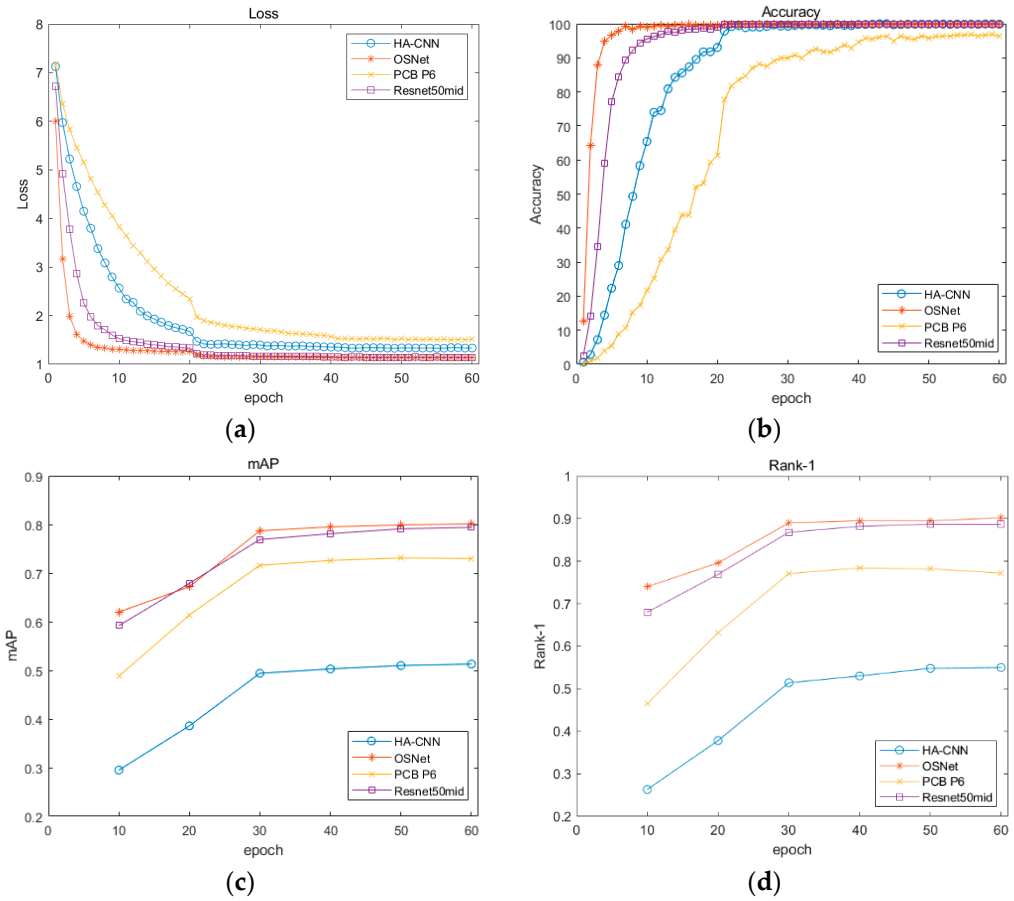


Figure 12. Performance evaluation of the re-identification models: (**a**) loss in training step; (**b**) accuracy in training step; (**c**) mAP in testing step; (**d**) Rank-1 in testing step.

5. Methods for Improvement of Vehicle Re-Identification

5.1. Reranking Method

Vehicle features can be extracted by the OSNet, and the Euclidean distance can be calculated by Equation (12)

$$d(p,q) = \| f_p - f_q \|_2^2 \tag{12}$$

where p and q stand for two vehicle images, and f_p, f_q represent their features, respectively. We apply the inverse as an indicator.

A reranking method presented by Zhong et al. [49] was adopted in this step. Reranking is a post-processing that helps to improve the initial ranking result without requiring extra labels or training data [50].

$$N(p,k) = \{g_1, g_2, g_3, ..., g_k\}, |N(p,k)| = k \tag{13}$$

As Equation (13) shows, we draw a set of pictures $\{g_1, g_2, g_3, \ldots, g_k\}$ as the k-nearest neighbors $N(p, k)$. It presents the initial order of Euclidean distances between gallery pictures and the query picture p. Then, we apply Equation (14) to obtain the k-reciprocal nearest neighbors $R(p, k)$, a subset of $N(p, k)$, whose members are more related to the query picture p and consider p as their k-nearest neighbors.

$$R(p,k) = \{g_i | g_i \in N(p,k) \& p \in N(g_i,k)\}, |N(p,k)| = k \tag{14}$$

The set $R(p, k)$ is the bidirectional k-nearest neighbors and effectively excludes false-positive pictures. In addition, a method was adopted to improve recall rate by Equation (15):

$$\begin{array}{l} R*(p,k) = R(p,k) \bigcup R(1, \frac{1}{2}k) \\ \text{s.t.} |R(p,k) \bigcap R(1, \frac{1}{2}k)| \geq \frac{2}{3}|R(1, \frac{1}{2}k)| \\ \forall q \in R(p,k) \end{array} \tag{15}$$

where $R^*(p, k)$ consists more positive samples, which are k/2-nearest neighbors of candidates in $R(p, k)$ and are likely to represent the same vehicle as the query picture. The $|R(1, \frac{1}{2}k)|$ denotes the number of members in the set $R(1, \frac{1}{2}k)$.

A pairwise distance $d_J(p, g_i)$ called Jaccard Distance between query picture p and gallery g is calculated by Equation (16) when g_i belongs to $R^*(p, k)$ otherwise is set to 0. It is a new index that stands for the relationships of similarity between pictures.

$$V_{p,g_i} = e^{-d(p,g_i)} \; s.t. g_i \in R^*(p, g_i) \tag{16}$$

$$d_J(p, g_i) = 1 - \frac{\sum_{j=1}^{N} \min\left(V_{p,g_i}, VL_{g_i,g_j}\right)}{\sum_{j=1}^{N} \max\left(V_{p,g_i}, VL_{g_i,g_j}\right)} \tag{17}$$

$V_{p, gi}$ is the initial similarity, a numerical value that converted from the initial distance. The final distance is defined by

$$d^*(p, g_i) = (1 - \lambda) d_J(p, g_i) + \lambda \mathtt{d}(p, g_i) \tag{18}$$

where λ is a constant that balances the effect of the initial distance $d(p, g_i)$ and the Jaccard Distance $d_J(p, g_i)$. Finally, re-ranked distances that express more accurate similarities are obtained.

5.2. Methods to Reduce the Number of Candidates

In order to reduce the number of candidates, vehicle direction information was taken into consideration since a vehicle almost never turns around on bridges. Similarities between vehicles in different directions can be set to 0. Along with directions, time information was used in this study. A statistical approach was applied to draw time consumed from one monitoring area to another. The time was assumed to be normally distributed, and mean and variance are calculated so as to work out the threshold by function

$$t = \overline{x} + 3\sigma \tag{19}$$

where t represents a threshold, and $\overline{x}$, σ denotes mean and variance, respectively. Because of the effects combined with time and directions, the number of candidates can be reduced.

5.3. Hungary Algorithm to Solve the Assignment Problem

Since the similarities have been computed, the goal is to determine the optimum assignment that maximizes the possibilities, which equals minimizing its opposite number. An approach based on the Hungary algorithm was utilized to match vehicles.

The Hungary algorithm consists of four steps, firstly (i) obtain the minimum value for each row, and subtract it from all the elements in that row; then (ii) every element minus the minimum value in its column; then (iii) use a minimum number of horizontal and vertical lines to cover zeros in the result, and if the number is equal to the number of rows, the positions of zeros are equal to the assignment result; and, finally, (iv) find the smallest element without a line covering it, then subtract it from all the uncovered elements and add it to the elements which are covered twice. Repeat step 3.

Thanks to the Hungary algorithm, the assignment problem for vehicles is solved, and information regarding the same vehicle from different cameras can be merged and output.

6. Field Validation of the Proposed Re-Identification Method

The proposed vehicle identification method was verified with video frames from three surveillance cameras on Jiubao Bridge, Hangzhou, China. Figure 13 shows their positions and directions, and Figure 14 demonstrates their views of the same truck. Apparently, camera 1 and camera 2 have completely varying observation directions, and camera 3 shares a similar sight with camera 1. Three ten-minute monitoring videos were obtained from the above cameras for validation.

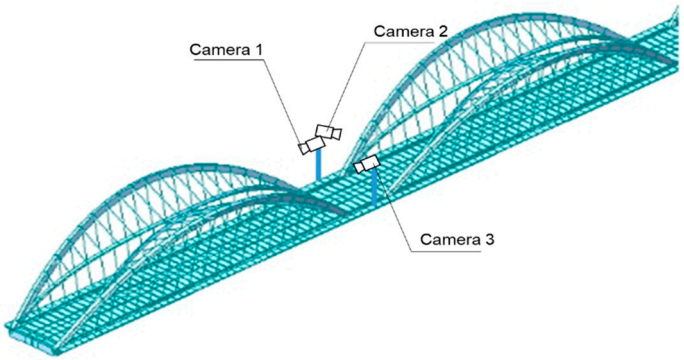


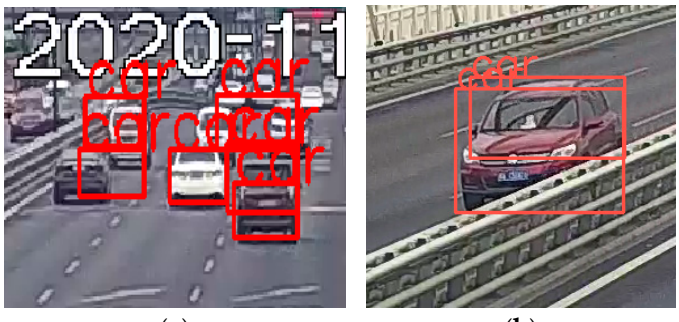
Figure 13. Layout of the cameras on the bridge.

(**a**) (**b**) (**c**)

Figure 14. Images of the same truck captured by three cameras: (**a**) camera 1; (**b**) camera 2; (**c**) camera 3.

6.1. Hungary Algorithm to Solve the Assignment Problem

After the detecting and tracking process, vehicle tracks and the temporal–spatial distribution of vehicles in the monitoring area were obtained. However, on account of false-positive detection, some wrong trajectory information was included. Figure 15a indicates that this algorithm failed to obtain a good performance in vehicle recognition, and this was due to the low resolution induced by the great distance between the camera and the vehicles. In addition, in a few cases, two bounding boxes or more were recognized to represent the same vehicle as illustrated in Figure 15b.



(**a**) (**b**)

Figure 15. Factors affecting recognition: (**a**) inexact positioning; (**b**) incorrect detection.

For precise tracking, two strategies were applied. The first one was to set up a monitoring area and only the vehicles detected in this area would be recorded. This strategy reduces monitoring range but improves robustness. The second one was to set a time limit to exclude the transitory track whose length is shorter. The limit of the time interval is 0.5 s in this study.

These operations ensure that only the correct vehicle trajectories will be stored. After post-processing, we applied accuracy defined by Equation (20) to denote its performance.

$$accuracy = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{20}$$

The TP refers to true positive detection, FP and FN refer to false positive detection and false negative detection, respectively. Although the proposed method is sensitive to the sights of cameras, it reached an accuracy over 97.4%, as shown in Table 3. Although the average accuracy is 97.7% which is satisfied, it is not as high as the accuracy achieved by Chen et al. [20] with SSD. The detection distance might be the main influencing factor that the visual field of this study (roughly 180 m) is much larger than that in their investigation. In addition, the performance of the SSD and the YOLO is also slightly different.

Table 3. Vehicle detection accuracy in video frames from three cameras.

Camera	Correct Detection	Mis-Detection	False Detection	Ground Truth	Accuracy
camera_1	909	14	2	923	98.3%
camera_2	905	23	1	928	97.4%
camera_3	909	7	17	916	97.4%
sum	2723	44	20	2767	97.7%

At this point, the tracking operations in a single camera were completed, and considering tracking through multiple non-overlapping cameras, another post-processing operation was used. A batch of pictures were captured based on the previous bounding box and the surveillance video. Taking computing time into account, the capturing operation was taken twice per second. Hence, every trajectory had its corresponding information regarding appearance.

6.2. Vehicle Re-Identification among Multiple Cameras

The first camera was set as the basic camera, the images of the same vehicle from different cameras were manually matched as the ground-truth for testing. Due to the difference of surveillance areas and missing detection, some vehicles only appear in one camera or two, especially at the beginning or the end of videos. Finally, 883 vehicles from camera 2 and 899 vehicles from camera 3 were found in camera 1.

The results of the field validation are summarized in Table 4. All the testing accuracies reach over 92.5% among the four testing groups that validates the effectiveness of the proposed vehicle identification method. In addition, the re-identification accuracy of test 2 is better than test 1 by 5.4% and the test 4 is better than test 3 by 5.2%. It is due to the fact that camera 1 and camera 3 have close direction for the visual field, which is opposite to that of camera 2. Moreover, seen from the comparison between test 1 and test 3, or test 2 and test 4, the utilization of re-ranking slightly improved the accuracy.

Table 4. Comparison among different cameras.

Test Number	Target Camera	Reranking or Not	Number of Vehicles	Correct Matches	False Matches	Accuracy
1	camera_2	N	883	817	66	92.5%
2	camera_3	N	899	881	18	97.9%
3	camera_2	Y	883	821	62	93.0%
4	camera_3	Y	899	883	16	98.2%

6.3. Recognition Results Based on Different Datasets

In order to investigate the vehicle re-identification performance of the proposed method, the public dataset, including the VeRi-776 and the VeRi-Wild, were utilized for testing. The former contains 51,038 images of 776 vehicles, among which 37,781 images were used for training and 13,257 images for testing. The VeRi-Wild contains 416,314 images of 40,671 vehicles, and 277,797 images were used for training and 138,517 images were for testing.

Table 5 shows their performances in field validation. Though VeRi-Wild has the largest amount of vehicle images, the model trained on it is not as good as the model trained on VeRi-776. Additionally, the models trained by the open-sourced datasets did not reach the same level as the model trained by the established dataset. The reasons responsible for their low accuracies are discussed in the next section.

Table 5. Comparison among different datasets.

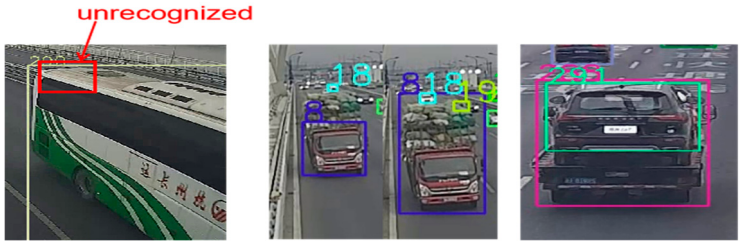
Test Number	Target Camera	Dataset	Number of Vehicles	True Matches	False Matches	Accuracy
1	camera_2	VeRi-776	883	526	357	59.6%
2	camera_3	VeRi-776	899	768	131	85.4%
3	camera_2	VeRi-Wild	883	297	586	33.6%
4	camera_3	VeRi-Wild	899	569	330	63.3%
5	camera_2	Jiubao	883	821	62	93.0%
6	camera_3	Jiubao	899	883	16	98.2%

7. Discussions of Incorrect Recognition

The discussion of errors helps open an insight into the issues for vehicle tracking along the whole bridge. In this section, the error discussion contains three parts, the vehicle detection errors in the visual field of the same camera, the vehicle re-identification errors among multiple cameras, and the reason for the different performances based on different datasets.

7.1. Recognition Results Based on Different Datasets

Figure 16 shows the main reasons that cause misdetection. The numbers in Figure 16 are the serial number of the detected vehicle and the bounding boxes were assigned with different colors to distinguish from each other. The blocked cars, shown in Figure 16a, should be responsible for false identification and this is the inherent defect of the vision-based method. The second one is the instability of detection, shown in Figure 16b. Despite the fact that the Kalman Filter was used to eliminate interference, the significant change in the bounding box can lead to mistakes. The last reason is the multiple boxes of the same object, as shown in Figure 16c. This usually does not last long and will be eliminated by our strategies, but when it happens, it can seriously affect vehicle tracking.



(**a**) (**b**) (**c**)

Figure 16. Reasons for misdetection: (**a**) blocked vehicles; (**b**) instability of detection; (**c**) multiple boxes of an object.

7.2. Discussions of Vehicle Re-Identification Errors

In the re-identification module, the input is the output of the detection module, so a query vehicle may not have its corresponding picture in the basic camera but will be forced to choose one, which means that misdetection will inevitably result in false matches. Another reason is the similar appearance of vehicles and samples are shown in Figure 17a. The low definition of surveillance cameras worsens this situation. Furthermore, the vehicles that are in close proximity with the query one will be captured and will interfere with feature extraction, as shown in Figure 17b.

(**a**) (**b**)

Figure 17. Samples of misidentification: (**a**) the similar appearance of different vehicles; (**b**) a picture interfered by another vehicle.

The results between camera 1 and camera 2 are not so desirable. The public dataset usually provides multi-views of vehicle images to restore the vehicle's appearance and applying multiple cameras that cover various sights as basic cameras may alleviate the problem. In view of the high accuracy between camera 1 and camera 3, a tentative means was applied to simulate multiple basic cameras in Table 6. In test 2, the pictures in camera 3 were relabeled as the label of their corresponding ones in camera 1 and mixed with that, while in test 3, they were input with no label and only worked in the reranking step.

Table 6. Comparison based on different basic cameras.

Test Number	Target Camera	Basic Camera	Re-Relabeling	Number of Vehicles	True Matches	False Matches	Accuracy
1	camera_2	1	/	883	821	62	93.0%
2	camera_2	1+3	Yes	883	852	31	96.5%
3	camera_2	1+3	No	883	843	34	95.5%

Though camera 3 does not share a similar sight with query pictures, it seems to benefit re-identification a lot if pictures are labeled. However, it is impractical to note pictures manually for industrial application. Moreover, accuracy has improved even if the additional pictures are unlabeled, which shows the effectiveness of reranking.

7.3. Discussions of the Difference of Performance among Multiple Datasets

The VeRi-Wild and VeRi-776 contain many more images than the established dataset but does not work as expected. The main reason is that their images have various definitions, as shown in Figure 18, which describes three typical images at the same dots per inch (DPI). It is obvious that VeRi-Wild has the highest definition and allows neural networks to recognize tiny parts, which does not work on our field validation due to the limitation of low definition.

Figure 18. Images in three datasets shown at the same DPI.

Figure 19 shows the heat maps of the same image obtained from diverse models and reveals another reason that the difference in distribution between training and testing data leads to a low accuracy. In the heat maps, the colors stand for the contribution of the corresponding areas to the identification of the vehicles. Darker color means larger contribution. The barriers and cables of the bridges which hardly appear in the VeRi-776 or VeRi-Wild seriously affect the recognition. When the models focus on the wrong object, wrong predictions are inevitable.

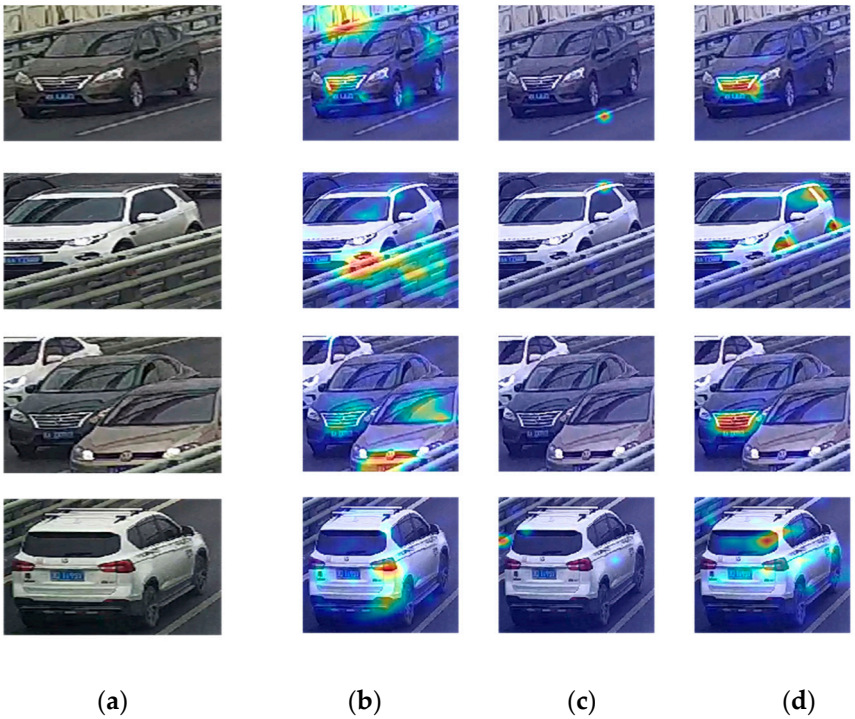


Figure 19. Heatmaps by models on various datasets: (**a**) original images; (**b**) VeRi-776; (**c**) VeRi-Wild; and (**d**) VeRi-Jiubao.

Through the above analysis, the significance of selecting an appropriate dataset similar to the object of study is clearly proved, and the distribution, along with the scale of the dataset, should be considered for better performance.

8. Conclusions

In this paper, a YOLOv4 and OSNet-based method for identification of temporal–spatial distribution of vehicle loads on bridges by means of multiple cameras was proposed, and a dataset containing 25,080 images of 1727 vehicles was established. Several re-identification models were adopted to conduct a comparison study with the OSNet, and a re-ranking method was applied to improve performances. Field validation on Jiubao bridge was conducted to verify accuracy of the proposed method. According to the study, some conclusions can be drawn as follows:

(1) The combination of the YOLOv4 and a modified IoU-based tracking method realizes the detection and tracking of vehicles on bridges in a single camera, and has an accuracy of 97.7%. In addition, the proposed adaptive lane recognition algorithm improves the location of vehicles precisely without extra considerable computation.
(2) In terms of the mAP and Rank-1 indices, the OSNet outperforms the other re-identification models and was chosen to verify our method. The accuracies of the OSNet-based re-identification method in field validation reached over 92.5% and 97.9% for camera 2 and camera 3, respectively, which indicates that vehicles can be precisely re-identified through multiple cameras without overlapped visual fields.
(3) With the introduction of the re-ranking method, the improvement in accuracy is 0.5% and 0.3% in camera 2 and camera 3, respectively. Though it only benefits the result slightly, further investigation shows that the effect can be enhanced by inputting more images even if they are unlabeled. The re-ranking method can reduce mistakes in vehicle re-identification, especially between two cameras with different sights.
(4) The realization of the proposed method can contribute to the acquisition of the temporal–spatial distribution of vehicles on the whole bridge for precise estimation of vehicle loads.

Author Contributions: Conceptualization, T.J. and X.Y.; methodology, T.J. and X.Y.; validation, T.J., X.Y. and Z.L.; formal analysis, Z.H.; investigation, T.J. and Z.H.; resources, X.Y. and Z.L.; data curation, Z.L.; writing—original draft preparation, T.J. and Z.L.; writing—review and editing, Z.H.; visualization, X.Y.; supervision, X.Y.; project administration, T.J. and X.Y.; funding acquisition, T.J. and X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: The work described in this paper was jointly supported by the China Postdoctoral Science Foundation (Grant No. 2022M712787), and the National Natural Science Foundation of China (Grant No. 52178306).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Szurgott, P.; Wekezer, J.; Kwasniewski, L.; Siervogel, J.; Ansley, M. Experimental Assessment of Dynamic Responses Induced in Concrete Bridges by Permit Vehicles. *J. Bridge Eng.* **2011**, *16*, 108–116. [CrossRef]
2. Yin, X.; Cai, C.S.; Liu, Y.; Fang, Z. Experimental and Numerical Studies of Nonstationary Random Vibrations for a High-Pier Bridge under Vehicular Loads. *J. Bridge Eng.* **2013**, *18*, 1005–1020. [CrossRef]
3. Zaurin, R.; Khuc, T.; Catbas, F.N. Hybrid Sensor-Camera Monitoring for Damage Detection: Case Study of a Real Bridge. *J. Bridge Eng.* **2016**, *21*, 05016002. [CrossRef]
4. Ye, X.W.; Jin, T.; Li, Z.X.; Ma, S.Y.; Ding, Y.; Ou, Y.H. Structural Crack Detection from Benchmark Data Sets Using Pruned Fully Convolutional Networks. *J. Struct. Eng.* **2021**, *147*, 04721008. [CrossRef]
5. Westgate, R.; Koo, K.-Y.; Brownjohn, J.; List, D. Suspension Bridge Response Due to Extreme Vehicle Loads. *Struct. Infrastruct. Eng.* **2014**, *10*, 821–833. [CrossRef]
6. Ye, X.W.; Li, Z.X.; Jin, T. Smartphone-Based Structural Crack Detection Using Pruned Fully Convolutional Networks and Edge Computing. *Smart Struct. Syst.* **2022**, *29*, 141–151. [CrossRef]

7. Fred Moses Weigh-in-Motion System Using Instrumented Bridges. *Transp. Eng. J. ASCE* **1979**, *105*, 233–249. [CrossRef]
8. Cantero, D.; González, A. Bridge Damage Detection Using Weigh-in-Motion Technology. *J. Bridge Eng.* **2015**, *20*, 04014078. [CrossRef]
9. Ye, X.W.; Su, Y.H.; Xi, P.S.; Chen, B.; Han, J.P. Statistical Analysis and Probabilistic Modeling of WIM Monitoring Data of an Instrumented Arch Bridge. *Smart Struct. Syst.* **2016**, *17*, 1087–1105. [CrossRef]
10. Gonçalves, M.S.; Carraro, F.; Lopez, R.H. A B-WIM Algorithm Considering the Modeling of the Bridge Dynamic Response. *Eng. Struct.* **2021**, *228*, 111533. [CrossRef]
11. Wu, Y.; Deng, L.; He, W. BwimNet: A Novel Method for Identifying Moving Vehicles Utilizing a Modified Encoder-Decoder Architecture. *Sensors* **2020**, *20*, 7170. [CrossRef]
12. Zhou, Y.; Pei, Y.; Zhou, S.; Zhao, Y.; Hu, J.; Yi, W. Novel Methodology for Identifying the Weight of Moving Vehicles on Bridges Using Structural Response Pattern Extraction and Deep Learning Algorithms. *Measurement* **2021**, *168*, 108384. [CrossRef]
13. Nguyen, T.T.; Pham, X.D.; Song, J.H.; Jin, S.; Kim, D.; Jeon, J.W. Compensating Background for Noise Due to Camera Vibration in Uncalibrated-Camera-Based Vehicle Speed Measurement System. *IEEE Trans. Veh. Technol.* **2011**, *60*, 30–43. [CrossRef]
14. Barcellos, P.; Bouvié, C.; Escouto, F.L.; Scharcanski, J. A Novel Video Based System for Detecting and Counting Vehicles at User-Defined Virtual Loops. *Expert Syst. Appl.* **2015**, *42*, 1845–1856. [CrossRef]
15. Lydon, D.; Taylor, S.E.; Lydon, M.; del Rincon, J.M.; Hester, D. Development and testing of a composite system for bridge health monitoring utilising computer vision and deep learning. *Smart Struct. Syst. Int. J.* **2019**, *24*, 723–732.
16. Zhou, Y.; Pei, Y.; Li, Z.; Fang, L.; Zhao, Y.; Yi, W. Vehicle Weight Identification System for Spatiotemporal Load Distribution on Bridges Based on Non-Contact Machine Vision Technology and Deep Learning Algorithms. *Measurement* **2020**, *159*, 107801. [CrossRef]
17. Ge, L.; Dan, D.; Li, H. An Accurate and Robust Monitoring Method of Full-bridge Traffic Load Distribution Based on YOLO-v3 Machine Vision. *Struct. Control Health Monit.* **2020**, *27*, e2636. [CrossRef]
18. Chen, Z.; Feng, Y.; Zhang, Y.; Liu, J.; Zhu, C.; Chen, A. An Accurate and Convenient Method of Vehicle Spatiotemporal Distribution Recognition Based on Computer Vision. *Sensors* **2022**, *22*, 6437. [CrossRef]
19. Chen, Z.; Ellis, T.; Velastin, S.A. Vehicle Detection, Tracking and Classification in Urban Traffic. In Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, AK, USA, 16–19 September 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 951–956.
20. Chen, L.; Zhang, Z.; Peng, L. Fast Single Shot Multibox Detector and Its Application on Vehicle Counting System. *IET Intell. Transp. Syst.* **2018**, *12*, 1406–1413. [CrossRef]
21. Harikrishnan, P.M.; Thomas, A.; Gopi, V.P.; Palanisamy, P. Fast Approach for Moving Vehicle Localization and Bounding Box Estimation in Highway Traffic Videos. *Signal Image Video Process.* **2021**, *15*, 1041–1048. [CrossRef]
22. Zhang, B.; Zhou, L.; Zhang, J. A Methodology for Obtaining Spatiotemporal Information of the Vehicles on Bridges Based on Computer Vision. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 471–487. [CrossRef]
23. Chen, Z.; Li, H.; Bao, Y.; Li, N.; Jin, Y. Identification of Spatio-Temporal Distribution of Vehicle Loads on Long-Span Bridges Using Computer Vision Technology: Spatio-Temporal Distribution Identification of Vehicle Loads. *Struct. Control Health Monit.* **2016**, *23*, 517–534. [CrossRef]
24. Dan, D.; Ge, L.; Yan, X. Identification of Moving Loads Based on the Information Fusion of Weigh-in-Motion System and Multiple Camera Machine Vision. *Measurement* **2019**, *144*, 155–166. [CrossRef]
25. Wu, C.-W.; Liu, C.-T.; Chiang, C.-E.; Tu, W.-C.; Chien, S.-Y. Vehicle Re-Identification with the Space-Time Prior. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 121–1217.
26. Wang, H.; Hou, J.; Chen, N. A Survey of Vehicle Re-Identification Based on Deep Learning. *IEEE Access* **2019**, *7*, 172443–172469. [CrossRef]
27. Zhang, Y.; Xie, J.; Peng, J.; Li, H.; Huang, Y. A Deep Neural Network-Based Vehicle Re-Identification Method for Bridge Load Monitoring. *Adv. Struct. Eng.* **2021**, *24*, 3691–3706. [CrossRef]
28. Khorramshahi, P.; Kumar, A.; Peri, N.; Rambhatla, S.S.; Chen, J.-C.; Chellappa, R. A Dual-Path Model With Adaptive Attention for Vehicle Re-Identification. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27–28 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 6131–6140.
29. Zhou, K.; Yang, Y.; Cavallaro, A.; Xiang, T. Omni-Scale Feature Learning for Person Re-Identification. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27–28 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3701–3711.
30. Pan, Y.; Wang, D.; Dong, Y.; Peng, B. A Novel Vision-Based Framework for Identifying Dynamic Vehicle Loads on Long-Span Bridges: A Case Study of Jiangyin Bridge, China. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 10441–10457. [CrossRef]
31. Shen, Y.; Xiao, T.; Li, H.; Yi, S.; Wang, X. Learning Deep Neural Networks for Vehicle Re-ID with Visual-Spatio-Temporal Path Proposals. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 27–29 October 2017; IEEE: Piscataway, NJ, USA, 2019; pp. 1918–1927.
32. Liu, X.; Liu, W.; Mei, T.; Ma, H. A Deep Learning-Based Approach to Progressive Vehicle Re-Identification for Urban Surveillance. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9906, pp. 869–884, ISBN 978-3-319-46474-9.

33. Yang, L.; Luo, P.; Loy, C.C.; Tang, X. A Large-Scale Car Dataset for Fine-Grained Categorization and Verification. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 3973–3981.
34. Lou, Y.; Bai, Y.; Liu, J.; Wang, S.; Duan, L. VERI-Wild: A Large Dataset and a New Method for Vehicle Re-Identification in the Wild. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3230–3238.
35. Munkres, J. Algorithms for the Assignment and Transportation Problems. *J. Soc. Ind. Appl. Math.* **1957**, *5*, 32–38. [CrossRef]
36. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
37. Tang, X.; Song, H.; Wang, W.; Yang, Y. Vehicle Spatial Distribution and 3D Trajectory Extraction Algorithm in a Cross-Camera Traffic Scene. *Sensors* **2020**, *20*, 6517. [CrossRef]
38. Li, Y.; Wang, H.; Dang, L.M.; Nguyen, T.N.; Han, D.; Lee, A.; Jang, I.; Moon, H. A Deep Learning-Based Hybrid Framework for Object Detection and Recognition in Autonomous Driving. *IEEE Access Pract. Innov. Open Solut.* **2020**, *8*, 194228–194239. [CrossRef]
39. Wang, C.-Y.; Mark Liao, H.-Y.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1571–1580.
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
41. Yang, J.; Xie, X.; Yang, W. Effective Contexts for UAV Vehicle Detection. *IEEE Access* **2019**, *7*, 85042–85054. [CrossRef]
42. Paliwal, K.; Basu, A. A Speech Enhancement Method Based on Kalman Filtering. In Proceedings of the ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing, Dallas, TX, USA, 6–9 April 1987; Volume 12, pp. 177–180.
43. Zhang, B.; Zhang, J. A Traffic Surveillance System for Obtaining Comprehensive Information of the Passing Vehicles Based on Instance Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 7040–7055. [CrossRef]
44. Yu, Q.; Chang, X.; Song, Y.-Z.; Xiang, T.; Hospedales, T.M. The Devil Is in the Middle: Exploiting Mid-Level Representations for Cross-Domain Instance Matching. *arXiv* **2018**, arXiv:1711.08106.
45. Sun, Y.; Zheng, L.; Yang, Y.; Tian, Q.; Wang, S. Beyond Part Models: Person Retrieval with Refined Part Pooling (and A Strong Convolutional Baseline). In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11208, pp. 501–518, ISBN 978-3-030-01224-3.
46. Li, W.; Zhu, X.; Gong, S. Harmonious Attention Network for Person Re-Identification. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 2285–2294.
47. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.
48. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 815–823.
49. Zhong, Z.; Zheng, L.; Cao, D.; Li, S. Re-Ranking Person Re-Identification with k-Reciprocal Encoding. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 3652–3661.
50. Peng, J.; Wang, H.; Zhao, T.; Fu, X. Learning Multi-Region Features for Vehicle Re-Identification with Context-Based Ranking Method. *Neurocomputing* **2019**, *359*, 427–437. [CrossRef]