Vehicle weight identification system for spatiotemporal load distribution on bridges based on non-contact machine vision technology and deep learning algorithms

Yun Zhou, Yilin Pei, Ziwei Li, Liang Fang, Yu Zhao, Weijian Yi

Hunan Provincial Key Lab on Damage Diagnosis for Engineering Structures, Hunan University, Changsha, Hunan 410082, China
College of Civil Engineering, Hunan University, Changsha, Hunan 410082, China
Key Laboratory for Green & Advanced Civil Engineering Materials and Application Technology of Hunan Province, Hunan University, Changsha, Hunan 410082, China
College of Water Resource and Civil Engineering, Hunan Agricultural University, Changsha, Hunan 410128, China

Abstract

Accurate information regarding the weight of vehicle loads plays a significant role in maintaining the structural health of bridges. However, the only method currently available for ascertaining the weight of loads is the bridge weigh-in-motion (BWIM) system, which is not widely used because of the high cost of the large device involved. There is therefore a need to develop an effective, low-cost technology to ascertain vehicle loads in relation to spatiotemporal load distribution on long-span bridges. This paper proposes a non-contact vehicle identification methodology to distinguish a vehicle from its load based on machine vision technology and deep learning algorithms. The vehicle information (i.e., type, weight, position, and motion trajectory, etc.) is conveniently obtained from a roadside monitoring surveillance camera, while the axle-weight distribution interval for nine classified vehicle types is obtained from the statistical information of 8402 delivery vehicles from which the relationship between a unique vehicle type and the corresponding weight information is established. Meanwhile, a dataset containing 8624 vehicle images was established for training the deep convolutional neural network (DCNN), where nine rough-grained vehicle classifications were contained in order to enhance the generalizability of the network. Optimization analysis was conducted to improve the network accuracy in vehicle types identification. The position of vehicles can also be effectively detected by a faster region-based convolutional neural network (Faster R-CNN), where the pre-trained DCNN with 98.17% vehicle types classification accuracy is employed as the co-shared network layer to enhance computation efficiency. Utilizing the object detection results from the Faster R-CNN and utilizing a Kalman filter, a vehicle in motion could also be simultaneously real-time tracked by the monitoring video, while a graphical user interface (GUI) incorporated into the video camera enabled automatic identification. A post-processing module has been established based on the proposed method, and a field test was conducted to validate the reliability of the system.

1. Introduction

The influence of vehicle motion plays a significant role in bridge design. Abnormal driving behavior in terms of loads compromise the structural performance of a bridge, which shortens its structural life and may even lead to collapse [1]. However, with the rapid development of the transportation industry, it is difficult to effectively ascertain vehicle loads on a bridge using structural health monitoring (SHM) and the intelligent transportation system (ITS). The current identification technology mainly relies on the bridge weigh-in-motion (BWIM) system, which was first proposed by Moses in 1979 by employing strain signals to distinguish the vehicle information (e.g. motion speed, axle spacing, and weights, etc.) [2]. Moses’ algorithm has also been used as the basic framework for the modern commercial BWIM system, and a number of studies have been conducted to enhance the system’s performance. For example, in 2010 O’Brien et al. [3] combined BWIM monitoring data with Monte-Carlo simulation results to evaluate the risk of traffic density to bridge structures, then in 2013 they employed...
the shear strain signals measured by BWIM gauges to detect the axle information of operational vehicles [4]. Zhao et al. [5] used bridge influence lines calibrated by field test results to improve the accuracy of the BWIM system in vehicle motion speed, axle spacing and weight identification. He et al. [6] presented a novel BWIM system combined with the virtual simply supported beam (VSSB) method, so that the motion speed and axle spacing of passing vehicles on bridges could be directly identified with the weighing sensors. Lydon et al. [7] improved the identification accuracy of the BWIM system by utilizing fibre optic sensors (FOS) as monitoring gauges, and Lydon et al. [8] conducted a comprehensive review of recent developments in the BWIM technique. However, the extensive use of the BWIM system is not feasible due to expensive hardware facilities. Meanwhile, the operation and maintenance of the of sensor gauges was generally required fully or partly closed traffic. It is therefore necessary to explore an efficient vehicle information identification technique for bridges without the support of the BWIM system.

A number of studies have also focused on the identification of vehicle loads from structural responses. Most of the proposed methods can be summarized into four categories: the interpretation method I (IMI), the interpretation method II (IMII), the time domain method (TDM), and the frequency-time domain method (FTDM) [9]. However, since the methods are based on the basic beam theory, their application has been limited to one-dimensional load identification information [10]. In order to identify vehicle information in a two-dimensional bridge deck, it is essential to develop an effective technology to obtain vehicle load information in spatial distribution.

Machine vision technology provides a direct way to monitor vehicle loads on a whole bridge deck, and the technology has been generally adopted as a novel way to identify vehicle load information on bridges. Zaurin and Catbas [11] proposed a novel SHM method using computer vision technology for bridge systems. Their laboratory tests showed that vehicle models could be effectively classified, detected and tracked from the monitoring videos, and the unit influence line of the bridge model was identified with output responses measured by the contact sensors. A field test was conducted by Catabs et al. [12] to evaluate the viability of the proposed method to be used for actual bridges, and found that the bridge unit influence line concurred with the finite element (FE) analysis results. Ojio et al. [13] ascertained the position and axle spacing of actual vehicles from camera monitoring videos, and the motion state was successfully tracked by machine vision technology based on the Lukas-Kanade method. Chen et al. [14] proposed an identification method to distinguish the spatiotemporal information of vehicle loads on long-span bridges, the weight of vehicles was obtained by the BWIM system, and the real-time motion trajectory was tracked by background subtraction and particle filter technology. Khuc and Catbas [15] ascertained vehicle loads and bridge responses based on machine vision technology, and evaluation of structural performance, including damage detection and localization, was successfully achieved with the obtained unit influence surface. Dan et al. [16] proposed a method to identify the information of moving vehicles based on the BWIM system and monitoring cameras, from which effective monitoring for real-time identification of vehicle weight, position and motion trajectory was achieved. The aforementioned research used machine vision monitoring technology as an assistant for the BWIM system for vehicle weight identification, which limited its wider application due to its high cost.

Deep learning algorithms, especially the deep convolutional neural network (DCNN) has been widely studied by engineering scholars. In recent years, a series of studies have been conducted by utilizing the strong capability of DCNN in the image recognition domain, since it is capable of learning appropriate classification features automatically without the need for human handcrafted features. Cha et al. [17] achieved a machine vision-based technology by utilizing the DCNN to classify concrete cracks with approximately 98% identification accuracy. Gopalakrishnan et al. [18] established an automatic system for classifying concrete cracks with the DCNN trained by the ImageNet database. Wang et al. [19] proposed the method to classify brick damage by training DCNN with series of damage images. Chen and Jahanshahi [20] utilized the DCNN to distinguish cracks on the metallic surfaces of nuclear power plants. Atta and Jahanshahi [21] proposed an approach to classify corrosion on metallic surfaces based on the DCNN. In the vehicle information identification field, Chen et al. [22] proposed a novel vehicle detection method by utilizing the DCNN model combined with probabilistic neural networks, which could effectively ascertain the vehicle information from surveillance traffic videos. Hu et al. [23] employed the multi-task DCNN model to classify fine-grained vehicle types of small cars. Furthermore, extensive research has been conducted based on the ability of deep learning algorithms to detect objects with input images. Wang et al. [24] proposed a novel method to rapidly detect damage on the Great Wall based on the faster region-based convolutional neural network (Faster R-CNN). Luo et al. [25] developed an automatic system to detect 22 classes of construction-related objects and 17 types of construction activities in still site images by utilizing Faster R-CNN. Cheng and Wang [26] conducted research into sewer pipe damage detection with Faster R-CNN, with results showing that the proposed method was accurate and fast. Yu et al. [27] achieved fine-grained vehicle type classification and detection of small cars based on the Faster R-CNN and joint Bayesian networks. Zhang et al. [28] achieved real-time traffic density evaluation based on a deep learning algorithm. An unmanned aerial vehicle was utilized to monitor traffic flow, and a monitoring video was trained by Faster R-CNN to detect vehicle targets in the videos. Jian et al. [29] proposed a novel traffic identification methodology based on sensors installed on bridges. The type, weight, motion speed and time–spatial distribution condition could be effectively distinguished by utilizing a deep learning algorithm and machine vision technology. Furthermore, a comprehensive review of the applications of deep learning technology in the civil infrastructure field was conducted by Ye et al. [30]. From the foregoing studies it can be concluded that the deep learning method provides an effective approach to classifying, detecting and tracking vehicles from monitoring videos.

However, for classifying vehicle type there is no global unified standard so that each country has its own set of criteria suggested by different industry departments [31–36]. Therefore, various vehicle classification methods have been proposed by scholars over the years. Sun and Ritchie [37] classified vehicles into seven different types: car, mini-van, sports and station wagon (Class-1), the SUV and pickup (Class-2), the van and full-size pickup (Class-3), the limo (Class-4), the 2-axle truck (Class-5), the trailer and bus (Class-6), and the multi-axles truck (Class-7). Stark et al. [38] established a dataset with 1904 images to classify 14 types of small cars. Lin et al. [39] divided vehicles into six different types including sedan, SUV, crossover, hatchback, wagon, and pickup truck, and established a corresponding fine-grained 3D dataset (named as FG3DCar) with 300 images. Liang et al. [40] established a vehicle dataset with 1482 images, where the vehicles were classified into eight different types by brand (i.e., Buick, Nissan, Volkswagen, Audi, BWM, Chevrolet, Citroen, and Toyota). Zhen et al. [41] divided vehicles into six classifications: bus, microbus, mini-van, sedan, SUV, and truck, and established the ‘BIT-Vehicle’ dataset with 9850 images captured from traffic surveillance cameras. Hu et al. [42] divided vehicles into five classifications: car, bus, truck, motorcycle, and van, and established the ‘SYSU-Vehicle’ dataset with 5000 images. Morris and Trivedi [43] established an image
dataset with eight classified vehicle types: sedan, pickup, SUV, van, merged, bike, truck, and semi. Sochor et al. [44] classified 27 different types of small car and established a 63,750-image dataset named 'BoxCars' by capturing frames from surveillance cameras. It would appear that most related research has focused on fine-grain classification of small cars. However, it is essential to also consider the influence of heavy trucks when assessing the performance of bridges, which requires classifying such vehicles.

For a state-of-the-art vision-based mobile vehicle detection method, it is essential to achieve an effective and fully non-contact vehicle information identification system based on machine vision technology. To the best of the authors’ knowledge, there is no published research that refers to vision-based vehicle weight classification. The authors of this paper therefore used an established dataset to achieve vehicle classification and weight estimation simultaneously. In the study, a non-contact machine vision technology is proposed to detect vehicle loads on a bridge based on deep learning algorithms. The resulting DCNN model was trained by an established dataset of 8624 vehicle images, and a series of optimal analyses were conducted to improve the accuracy of the networks rough classification of vehicles. Therefore, the weight information of classified vehicle types could be predicted based on the established mapping relationship. The Faster R-CNN and Kalman filter was utilized to achieve the real-time detecting and tracking of target vehicles, respectively. Finally, an automatic identification system comprising a monitoring camera and post-processing software modules has been established based on the proposed method.

2. Framework of the proposed vehicles identification method

The global framework of the proposed system is presented in Fig. 1 below.

As a novel technology in the SHM of bridges, non-contact machine vision technology has the unique advantages of rapid implementation, easy operation, and global measurement, which are important for rapid bridge condition assessment. However, this technology has been negatively affected by the vehicle weight extraction process that relies on the support of high-cost BWIM systems. To address this problem, the Structural Health Monitoring Research Team in Hunan University (www.hnutest.com) undertook this study to achieve a fully automatic camera monitoring system, and finally to investigate the inversion of bridge influence surface based on non-contact machine vision monitoring technology and interval affine arithmetic, and also combining the advanced methodologies of deep learning theory, computer graphics, and interval algorithms. To achieve the vehicle information identification based on machine vision technology, this study focused on Research-1 as shown in Fig. 1 above.

In order to gather information about the moving vehicle loads (i.e., type, weight, and position, etc.) from the monitoring videos, the surveillance camera was arranged by the roadside along the bridge to record the traffic flow. To specifically evaluate the bridge performance influenced by vehicle loads, the novel vehicle classification standard was employed to achieve the rough-grained classification of different vehicle types. This allowed the mapping relationship between certain vehicle types and the corresponding

Fig. 1. Framework of the proposed vehicle load identification system. (Note: the red dotted windows indicate corresponding global research projects, and the black dotted windows denotes the corresponding sub-research of this study’s Research-1). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
weight distribution information to be built. An image dataset of the nine classified vehicles was then established for DCNN training, and a series of optimal training methods were employed to improve the vehicle types identification accuracy. Based on the distinguished vehicle types, the corresponding weight interval could also be directly predicted from the monitoring videos without the support of the BWIM system. Meanwhile, the Faster R-CNN was utilized to obtain the position of motion vehicles from the monitoring videos. And for vehicles visible in the camera monitoring region, after initializing the parameters of the Kalman filter, the target tracking strategy could be achieved through frame sequencing. Furthermore, an automatic software program was employed that comprised four modules: a vehicle weight prediction module, a vehicle type classification module, a vehicle object detection module, and a target vehicle tracking module. And the vehicle parameters in terms of type, weight, position, and driving trajectory could be directly distinguished from the monitoring video by the proposed machine vision technology.

3. Data preparation

3.1. Statistics for vehicle parameters

As one of the typical external loads on bridges, conducting effective monitoring technology to obtain vehicle load information (e.g., type, weight, position, and motion speed, etc.) plays a significant role in bridge performance assessment. Relying on the BWIM system has obvious shortcomings including high device cost and complex installation process. Although the novel identification methodology has been proposed based on machine vision technology, the vehicle information including the position, moving speed and the axle spacing could be clearly distinguished with the monitoring camera. However, the corresponding weight information cannot be achieved without the support of the BWIM system. It is therefore essential to achieve an effective and fully non-contact vehicle information identification system based on camera vision technology.

In order to directly obtain the weight of vehicles from monitoring videos, it is necessary to build a mapping relationship between the classified vehicle types with corresponding weight information. Therefore, rough-grained classification criteria were proposed to classify common vehicles into nine types (sedan car, mini bus, light truck, large truck, coach truck, 2-axle truck with 4 wheels, 2-axle truck with 6 wheels, 3-axle truck, and the 4-axle truck) with the axle spacing could be clearly distinguished with the monitoring camera. However, the corresponding weight information cannot be achieved without the support of the BWIM system. Therefore, the vehicle motion videos were recorded on the Houzishi Bridge that crosses the Xiangjiang River in Changsha, China, it is a continuous box girder bridge with V-shaped diagonal supports that has a main span of 1390 m. For capturing vehicles on the bridge, a Canon 6D monitoring camera was installed at the western bridge approach and set for recording vehicle motion videos with a resolution of 1920 × 1080 pixels. The camera recording frame rate was 25 fps and the video duration period was 90 min 54 s. The recorded vehicle motion video was transformed into sequence images by Matlab software, through which 125,000 images were captured in total. The raw frames were selected to ensure that each image only contained one complete vehicle object, and the identical image processing mode was also applied during the website collection process. Finally, an image dataset named HNU-Vehicle Dataset including 8624 images of nine vehicle types was established to achieve the DCNN training framework. The monitoring camera captured 1183 images while another 7441 images were collected from website, as shown in Fig. 3.

4. Vehicles classification identification

4.1. Overall network operations

The DCNN is a deep learning algorithm designed to extract relevant features of input data. The model consists of three types of network layer: the convolution (CONV) layer, the pooling (POOL) layer, and the fully connected (FC) layer [45]. For the CONV layer, the multiplication operation is conducted for each parameter in the convolution (CONV) layer, the pooling (POOL) layer, and the fully connected (FC) layer [45]. The multiplicative-obtained value is then added to the bias to acquire the final output elements. The processing details of the convolution operations is shown in Fig. 4(a); the input data can be effectively reduced at this stage to enhance calculation efficiency. Meanwhile, the ‘stride’ parameter represents the amount of the receptive field's pixels scanned above the input array in the horizontal and vertical directions. And it should be noticed that although the calculation efficiency would be enhanced with a larger stride size, the acquired features of the input data would be lost under the corresponding condition. Furthermore, the size of array would be reduced again when input to the POOL layer under the corresponding pooling operation. The max pooling operation would select the maximum value of scanned sub-arrays, while the
mean pooling operation usually adopts the calculated average value, as shown in Fig. 4(b). Meanwhile, related studies have indicated that image dataset analysis under max pooling would be better than under mean pooling [46]. The features of raw input data would be extracted and transformed into higher-level expression forms after a series of convolution and pooling operations, and the feature classification function would be achieved by the FC layers. Generally, the FC layers are designed at the bottom of global DCNN architecture, and the last FC layer is combined with the softmax function to achieve output classifications. The classification process is represented by the probabilistic expression as shown in Eq. (1):

Table 1
Distribution interval of vehicle parameters.

<table>
<thead>
<tr>
<th>Vehicle type</th>
<th>Axle spacing (mm)</th>
<th>Global weight (t)</th>
<th>Front-axle weight (t)</th>
<th>Rear-axle weight (t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sedan car</td>
<td>[2550, 3150]</td>
<td>[1.00, 3.00]</td>
<td>[0.50, 1.40]</td>
<td>[0.50, 1.80]</td>
</tr>
<tr>
<td>Mini bus</td>
<td>[1900, 4400]</td>
<td>[0.95, 5.50]</td>
<td>[0.35, 2.00]</td>
<td>[0.50, 2.85]</td>
</tr>
<tr>
<td>Light truck</td>
<td>[3100, 4100]</td>
<td>[2.00, 6.50]</td>
<td>[0.50, 2.70]</td>
<td>[0.50, 4.20]</td>
</tr>
<tr>
<td>Large truck</td>
<td>[3800, 6300]</td>
<td>[1.40, 18.00]</td>
<td>[0.70, 6.50]</td>
<td>[0.70, 11.50]</td>
</tr>
<tr>
<td>Coach truck</td>
<td>[3300, 6100]</td>
<td>[6.00, 18.00]</td>
<td>[2.10, 6.50]</td>
<td>[3.30, 11.50]</td>
</tr>
<tr>
<td>2A-4 W truck</td>
<td>[2000, 3450]</td>
<td>[0.70, 3.50]</td>
<td>[0.35, 1.80]</td>
<td>[0.35, 2.25]</td>
</tr>
<tr>
<td>2A-6 W truck</td>
<td>[2300, 5800]</td>
<td>[1.40, 18.00]</td>
<td>[0.50, 7.00]</td>
<td>[0.90, 11.50]</td>
</tr>
<tr>
<td>3A truck</td>
<td>[1300, 5800]</td>
<td>[7.30, 25.00]</td>
<td>[1.60, 7.50]</td>
<td>[3.10, 18.00]</td>
</tr>
<tr>
<td>4A truck</td>
<td>[1500, 2100]</td>
<td>[8.80, 32.00]</td>
<td>[2.00, 7.50]</td>
<td>[3.50, 9.00]</td>
</tr>
</tbody>
</table>

Note: the 2A-4 W truck and 2A-6 W truck indicates the 2-axle truck with 4 and 6 wheels, respectively. And the 3A truck and 4A truck indicates the 3-axle truck and 4-axle truck, respectively.

Fig. 2. Statistical parameters of two vehicle types: (a) sedan car; (b) 2-axle truck with 4 wheels. (Note: CW denotes the curb weight, MPTW denotes the maximum permissible total weight, AS denotes the axle spacing, FAWFL denotes the front-axle weight under fully loaded state, FAWUL denotes the front-axle weight under unloaded state, RAWFL denotes the rear-axle weight under fully loaded state, and RAWUL denotes the rear-axle weight under unloaded state).
contains 356 kernels with a 5 × 5 × 96 field size employing max pooling operations. Additionally, the local response normalization (LRN) operation was imported to deal with the pooling results of the first two CONV layers. The 3rd CONV layer contains 38 kernels employing max pooling operation. Furthermore, the dropout technique was implemented to prevent the overfitting phenomenon, which resulted in a more robust DCNN model.

The training process of the DCNN model was conducted with Matlab software on a computer workstation with the following specifications: Intel(R) i7-8700 @3.20 GHz CPU; 16 GB of RAM; and a GTX 1060Ti GPU). The stochastic gradient descent momentum (SGDM) algorithm [48] was utilized to update the weights of the initialized AlexNet model with training parameters of 0.9 momentum and 5 × 10^{-4} decay. The global training process was conducted for an epoch batch size of 100, and a maximum of 300 of iterations. Parameter analysis was conducted by defining the base learning rate for 1 × 10^{-4}, 1 × 10^{-3} and 1 × 10^{-2}. A total of 8624 image datasets were utilized for the AlexNet model training process, with 75% (6467 datasets) randomly selected for training and the remaining 25% (2157 datasets) used for testing. Details of the training and testing dataset for the nine classified vehicle types is summarized in Table 3. To satisfy the specific requirements of image pixel size for AlexNet model inputs, each of the

### Table 2
Details of the open-sourced vehicle images dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>Description</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>KITTI</td>
<td>389 stereoscopic vehicle</td>
<td>Dataset for stereo vehicle detection</td>
<td><a href="http://www.cvlibs.net/datasets/kitti/eval_object.php">http://www.cvlibs.net/datasets/kitti/eval_object.php</a></td>
</tr>
<tr>
<td>BDD100K</td>
<td>100 thousand images with</td>
<td>Open-sourced dataset for autonomous driving</td>
<td><a href="https://bair.berkeley.edu/blog/2018/05/30/bdd">https://bair.berkeley.edu/blog/2018/05/30/bdd</a></td>
</tr>
<tr>
<td>TME Motorway Dataset</td>
<td>30 thousand images with type label</td>
<td>Captured by vehicle driving recorder</td>
<td><a href="http://cmp.felk.cvut.cz/data/motorway">http://cmp.felk.cvut.cz/data/motorway</a></td>
</tr>
<tr>
<td>Cars Dataset</td>
<td>16 thousand images including 196 vehicle types</td>
<td>Focus on various types of sedan car</td>
<td><a href="http://ai.stanford.edu/~j">http://ai.stanford.edu/~j</a> krause/cars/car_dataset.html</td>
</tr>
<tr>
<td>CompCars</td>
<td>136 thousand images including 1716 vehicle types</td>
<td>Captured by traffic monitoring cameras</td>
<td><a href="http://mmlab.ie.cuhk.edu.hk/datasets/compCars">http://mmlab.ie.cuhk.edu.hk/datasets/compCars</a></td>
</tr>
</tbody>
</table>

![Fig. 3. Examples of the nine classified vehicle types in the established dataset.](image)
established image datasets with various sizes were all scaled into 227 x 227 pixels for the AlexNet model training process. The classified image datasets with corresponding labels were input to the AlexNet model.

The AlexNet training process on the input dataset is shown in Fig. 6(a) and (b). It can be seen that when the base learning rate is defined as $1 \times 10^{-4}$, the training accuracy curve generally remains stable at around 40% at the 40th epoch, while the function loss curve stabilizes around 1.6 at the 40th epoch. The global training time was approximately about 2 h 19 min, and the average precision (AP) value on test datasets was observed to be only 25.81%. However, when the base learning rate was defined as $1 \times 10^{-3}$, the training accuracy curve and loss function curve generally stabilized around 80% at the 40th epoch and 0.5 at the 50th epoch, respectively. The global training time was approximately 2 h 11 min, while the AP value significantly reached 72.29%. Furthermore, when the base learning rate was defined as $1 \times 10^{-2}$, the training accuracy curve and loss function curve generally stabilized around 99% at the 40th epoch and 0.03 at the 40th epoch, respectively. The global training time was approximately 2 h 20 min, and the highest AP value was 86.49%. The corresponding test results expressed by the confusion matrix, which was utilized to evaluate the accuracy of the image classification, is presented in Fig. 6(c). From the figure, it can be seen that the sedan car has the highest accuracy of 100%, while the accuracy of a 2-axle truck with 6 wheels is the lowest.

### Table 3
Details of the DCNN training set.

<table>
<thead>
<tr>
<th>Vehicle type</th>
<th>Training set (n)</th>
<th>Testing set (n)</th>
<th>Total amounts (n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sedan car</td>
<td>626</td>
<td>207</td>
<td>833</td>
</tr>
<tr>
<td>Mini bus</td>
<td>615</td>
<td>209</td>
<td>824</td>
</tr>
<tr>
<td>Light truck</td>
<td>446</td>
<td>147</td>
<td>593</td>
</tr>
<tr>
<td>Large truck</td>
<td>130</td>
<td>44</td>
<td>174</td>
</tr>
<tr>
<td>Coach truck</td>
<td>441</td>
<td>147</td>
<td>588</td>
</tr>
<tr>
<td>2A-4 W truck</td>
<td>636</td>
<td>212</td>
<td>848</td>
</tr>
<tr>
<td>2A-6 W truck</td>
<td>1202</td>
<td>400</td>
<td>1602</td>
</tr>
<tr>
<td>3A truck</td>
<td>1298</td>
<td>433</td>
<td>1731</td>
</tr>
<tr>
<td>4A truck</td>
<td>1073</td>
<td>358</td>
<td>1431</td>
</tr>
<tr>
<td>Total</td>
<td>6467</td>
<td>2157</td>
<td>8624</td>
</tr>
</tbody>
</table>

![Fig. 4. Convolution and pooling operations: (a) convolution operation; (b) pooling operation. (Note: the red and blue windows denote the sub-arrays scanned across the global input array in vertical and horizontal directions. And the solid and dotted window denotes the current and following sub-array respectively). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)](image)

![Fig. 5. Architecture of the AlexNet model.](image)
with an accuracy of only 74.00%. According to the test results, the sedan car, large truck, coach truck, and 2-axle truck with 4 wheels distinguish themselves with an accuracy greater than 90%, while the accuracy of mini bus, 2-axle truck with 6 wheels, 3-axle truck, and 4-axle truck are lower than the mean accuracy. The reason for these mis-classifications can be summarized as follows: (1) the amount of training set for light truck was relatively less than that of other vehicle types, which might cause the trained network to not effectively extract the specific features that distinguish a mini bus from a light truck; (2) the specific features in terms of appearance and dimensions were basically identical for the 2-axle truck with 6 wheels, the 3-axle truck, and the 4-axle truck, and the only distinguishing feature that could be extracted by the DCNN might be the number of wheels.

4.3. Optimization training for AlexNet model

Since approximately 60 million parameters are involved in the AlexNet model training process, the limited dataset used for the DCNN model might limit its ability to learn. However, the transfer learning proposed by Pan and Yang [49] provides a powerful solution for training a DCNN with a limited dataset. Typically, the generic features (i.e., edge, and colour blobs, etc.) extracted by the DCNN are generally similar among different image datasets. Generic features with strong transferability are mostly learned by the lower-level CONV layers in the DCNN, while the specific features of the target dataset are learned by the corresponding higher-level layers. Therefore, the knowledge learned from a task can be used to deal with subsequent tasks by initializing the parameters of higher-level layers of a pre-trained DCNN model, while the corresponding information of lower-level CONV layers is reserved to accelerate the global training process, as shown in Fig. 7. Specifically, the transfer learning mode is extremely useful in big-data-driven learning regions such as deep learning.

To achieve high performance, the transfer learning results of the AlexNet model pre-trained by the ImageNet with 14 million images collected from over 1000 object classes [50] were utilized in the vehicle types classification task. Corresponding information of CONV layers were sustained but only the parameters of the bottom FC layers were initialized, and a softmax nonlinear activation function was combined to serve as an output label classifier. During the training process, the SGD algorithm was utilized with the training parameters of 0.9 momentum and $5 \times 10^{-4}$ decay, and the batch size, maximum iteration, and base learning rate was defined as 32, $1 \times 10^2$, and $1 \times 10^{-3}$, respectively. In this stage, the global training time was significantly reduced to 36 min based on the same computer workstation and Matlab platform, and the highest AP value for 87.68% was acquired through transfer learning from the ImageNet dataset. The predicted results are expressed by the confusion matrix as shown in Fig. 8. Compared with the transfer learning results of the former direct learning mode, it was found that the computation efficiency was enhanced approximately 5 times by preserving the extracted generic features of low-level CONV layers obtained from the ImageNet dataset. However, the AP value was not improved in the pre-trained AlexNet model.
To further improve the performance of the pre-trained AlexNet model, a data augmentation function was employed to increase the image amounts in the training stage. Raw images of established vehicle datasets have been calculated by a series of image processing techniques (i.e., random flipping, rotation, translation and scaling) in each training epoch, as shown in Fig. 9, and the existing processed images were summed up and transformed as inputs to the next training epoch.

The strengthening learning strategy was conducted by the pre-trained AlexNet model, with the batch size, maximum iteration number, and base learning rate defined as 32, $1 \times 10^2$, and $1 \times 10^{-3}$, respectively. Meanwhile, a parameter study was conducted to reveal the influence of image processing levels on network performance. Details of the defined image processing quantity in each training case are listed in Table 4 along with corresponding AP values of the trained network. It can be seen that the AP of defined cases were both significantly enhanced to 90%, with the highest performance of 93.46% in Case-2. The corresponding predicted results are shown in Fig. 10, from which it is found that the test accuracy for most vehicle types is over 90%; the exceptions being the 3-axle and 4-axle trucks that were enhanced to 84% and 80% respectively. The results indicate that the strengthening learning strategy using image processing techniques can ensure that the DCNN model fully extracts features from a limited input dataset. This significantly enhances the generalizability of networks to avoid the over-fitting phenomenon, especially without the necessity to manually increase the dataset amounts.
4.4. VGG-16 model training

Due to the prediction accuracy of 3-axle and 4-axle trucks needing to be improved, the VGG-16 model with its excellent performance in image classification and a deeper network architecture [51], was chosen for this study. The architecture of the VGG-16 model is shown in Fig. 11, which is composed of CONV layers, POOL layers, and FC layers. It should be noted that small convolution kernels rather than large ones are utilized in VGG-16 compared with the AlexNet, leading to the total number of parameters reaching approximately 120,000,000. VGG-16 is therefore widely recognized as the DCNN model with the best generative capacity.

The global training task was conducted based on the strengthening training strategy for AlexNet mentioned above. The features of low-level CONV layers extracted from ImageNet were sustained but only initialized the parameters of the bottom FC layers, and a data augmentation function was also employed to increase the number of images used in the training stage. Parameters such as batch size, maximum iteration, and base learning rate were defined as 32, 60, and $10^{-3}$ respectively, and the image processing technology remained identical to that of Case-2 in Table 4. The vehicle images of established dataset were resized to $224 \times 224$ pixels blocks for the network training process, and a training result of 98.17% AP value was eventually obtained for the VGG-16 model, as shown in Fig. 12. The figure shows that

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Random flipping</th>
<th>Rotation angle (°)</th>
<th>Horizontal translation (pixels)</th>
<th>Vertical translation (pixels)</th>
<th>Horizontal scaling (pixels)</th>
<th>Vertical scaling (pixels)</th>
<th>AP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>[-20, 20]</td>
<td>[-10, 10]</td>
<td>[-10, 10]</td>
<td>[0.8, 1.2]</td>
<td>[0.8, 1.2]</td>
<td>92.55</td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>[-20, 20]</td>
<td>[-10, 10]</td>
<td>[-10, 10]</td>
<td>[0.9, 1.1]</td>
<td>[0.9, 1.1]</td>
<td>93.46</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>[-20, 20]</td>
<td>[-5, 5]</td>
<td>[-5, 5]</td>
<td>[0.8, 1.2]</td>
<td>[0.8, 1.2]</td>
<td>91.76</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>[-20, 20]</td>
<td>[-5, 5]</td>
<td>[-5, 5]</td>
<td>[0.9, 1.1]</td>
<td>[0.9, 1.1]</td>
<td>92.14</td>
</tr>
<tr>
<td>5</td>
<td>Yes</td>
<td>[-15, 15]</td>
<td>[-10, 10]</td>
<td>[-10, 10]</td>
<td>[0.8, 1.2]</td>
<td>[0.8, 1.2]</td>
<td>91.05</td>
</tr>
<tr>
<td>6</td>
<td>Yes</td>
<td>[-15, 15]</td>
<td>[-10, 10]</td>
<td>[-10, 10]</td>
<td>[0.9, 1.1]</td>
<td>[0.9, 1.1]</td>
<td>91.55</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>[-15, 15]</td>
<td>[-5, 5]</td>
<td>[-5, 5]</td>
<td>[0.8, 1.2]</td>
<td>[0.8, 1.2]</td>
<td>90.83</td>
</tr>
<tr>
<td>8</td>
<td>Yes</td>
<td>[-15, 15]</td>
<td>[-5, 5]</td>
<td>[-5, 5]</td>
<td>[0.9, 1.1]</td>
<td>[0.9, 1.1]</td>
<td>91.22</td>
</tr>
</tbody>
</table>

Fig. 10. Strengthen learning results of the AlexNet framework.

Fig. 11. Architecture of the VGG-16 model.
the prediction accuracy for nine classified vehicle types all reached above 96%, with the 3-axle and 4-axle truck enhanced to 96.3% and 99.4% and 100% accuracy achieved for the sedan car, mini bus, and light truck. This indicates that the DCNN performance has been strengthened by increasing the amount of network layers. Therefore, moving vehicles can be distinguished by using cameras with the trained VGG-16 model.

5. Identification test for vehicle detection

5.1. Architecture of Faster R-CNN

As a typical multi-task network for target localization and classifying functions, the Faster R-CNN has been widely utilized in the target detection field and is composed of two deep networks with co-shared convolutional layers - the region proposal network (RPN) and the Fast R-CNN [52]. The input image was firstly transferred into the RPN, where the specific network was end-to-end trained by the SGD with a learning rate of global Faster R-CNN. Furthermore, the generalizability of the CNN and the RPN, which play a significant role in the performance of the trained VGG-16 model.

Fig. 12. Strengthen learning results of VGG-16 training framework.

5.2. Data processing

In order to improve the generalizability of the Faster R-CNN model, the same dataset of 8642 images mentioned above, including nine classified vehicle types, was utilized for the network training process. Meanwhile, the trained VGG-16 model with a 98.17% AP value in vehicle type identification was selected as the co-shared CONV layer between the RPN and Fast R-CNN. The vehicle images training set randomly selected from the dataset was input to the Faster R-CNN framework, and the vehicle in each of the raw images was selected by a manually annotated rectangular box.
Specifically, the complete target vehicle should be selected in the rectangle box, while the corresponding background information should be obtained since it is important for predicting the vehicles in the following steps. However, due to the length of the rectangular box should be selected which was less than 32 pixels, the images with small vehicle target should be ignored during the processing stage. The XML format file was then generated by the labelled dataset, which was expressed as the \([x, y, w, h] / \text{C138}\) form including the related information as central coordinates, width and height of selected target region, respectively.

5.3. Testing of vehicle detection

The vehicle detection task was conducted by preliminarily initializing the corresponding parameters of the Faster R-CNN. Each anchor was set with three various aspect ratios as 2:1, 1:1, or 1:2, and each anchor defined with three different sizes as 128, 256, or 512. Therefore, a total of 9 different types of anchors could be obtained during the Faster R-CNN training process. The selected dataset with corresponding XML format file were both input to the network, and the alternating training algorithm was employed in the training process of the Faster R-CNN network, which included four steps. The first step was to initialize the RPN with the pre-trained VGG-16 model using the input vehicle image dataset. Secondly, the initialized Faster R-CNN was trained by using the region proposals generated by the preceding RPN from Step 1. The parameters of the CONV layers were then fixed to co-share information between the RPN and the Fast R-CNN model, and the RPN was trained again in this step. Finally, the Fast R-CNN model was trained once again by the generated region proposals from Step 3.

A series of region proposals would be generated with one target object, and the optimal region that contains the highest overlap proportion between the predicted position and the actual position would be selected by utilizing the non-maximum suppression (NMS) algorithm [53]. The intersection-over-Union (IoU) overlap ratio, which is an evaluation metric used to measure the accuracy of an object detector on a particular dataset, was defined as 0.6 in this study. This indicates that the predicted region with an IoU overlap ratio higher than 0.6 would be distinguished as a positive label. Meanwhile, the SGDM algorithm was employed with a momentum of 0.9 in the test, and the base learning rate and weight decay of the Faster R-CNN model was defined as \(1 \times 10^{-3}\) and \(5 \times 10^{-4}\), respectively. The 26-hour global training process was conducted by Matlab software based on a computer equipped with an Intel(R) i7-8700 CPU @3.20 GHz, 16 GB of RAM, and a GTX 1060Ti GPU.

Vehicle images with various test conditions (i.e., types, target dimensions, and lighting conditions, etc.) have been utilized to...
evaluate the performance of the trained Faster R-CNN. The results indicate that the target vehicle could be effectively distinguished with accurate identification and position information, as shown in Fig. 14. Therefore, an operational vehicle could be well detected by machine vision with the trained Faster R-CNN model.

6. Real-time tracking of vehicles in motion

6.1. Kalman filter tracking theory

In order to realize the spatiotemporal localization of vehicle loads automatically from monitoring videos, it is essential to conduct research on the real-time tracking of moving vehicles. As a type of recursive filtering method, the Kalman filter tracking algorithm employs the Bayesian theory and the linear minimum variance theory to build the object motion estimation model, which could be utilized to acquire the real-time spatiotemporal position of a target object [54]. In this section, the Kalman filter method is employed to conduct vehicle tracking research.

Generally, the Kalman filter tracking algorithm consists of a state prediction model calculated by the object state transition function, and the state update model calculated by the object state measurement function. The state variable of the present moment would be predicted by the detected target at the last moment, and after the features extracting process at the neighbouring region around the predicted position, the measurement vector would be obtained and then utilized to iteratively update the state vector until the optimal state estimation of present moment is achieved. The object state transition function $K_i$ can be described as follows [55]:

$$K_i = AK_{i-1} + \xi_{i-1}$$

where $A$ denotes the transition matrix transferred from last moment to present moment.

Meanwhile, the object state measurement function $L_i$ can be calculated as follows:

$$L_i = HK_{i-1} + \eta_i$$

where $H$ denotes the measurement matrix. $\xi_{i-1}$ and $\eta_i$ denotes the systematic and observation Gaussian noise following a normal distribution, respectively.

However, due to vector $K_i$ not being directly measured, the states at the present time, including state variables and covariance errors, are both estimated by the motion state of the target object in the next moment; the measurement state $L_i$ is utilized to update the unknown state $K_i$. The predicted equation of the state variables at a given moment can be expressed as follows:

$$\bar{K}_i = AK_{i-1} + \xi_{i-1}$$

And the predicted equation of the covariance function errors at a given moment can be expressed as follows:

$$P_i = AP_{i-1}A^T + Q_i$$

where $Q_i$ denotes the covariance function of systematic Gaussian noise.

The state updating model includes the updating process to Kalman weights and the optimizing process to state variables and covariance errors. The gain equation of Kalman weights can be expressed as follows:

$$\bar{K}_i = K_i + \omega_i (L_i - HK_i)$$

Furthermore, the optimization equation utilized for updating the covariance errors can be obtained as follows:

$$P_i = (1 - \omega_i H)P_i$$

Fig. 15. Flowchart of vehicle tracking process.
In the actual vehicle monitoring videos, the position of target vehicles in the next frame could be obtained by the Kalman filter once the state transition function and state measurement function are determined, and the motion trajectories of the target object obtained [56].

6.2. Testing of vehicle tracking

To achieve real-time tracking of vehicles in motion using recorded videos, the corresponding parameters in the Kalman filter basic equations need to be initialized when the target vehicle is visible in the frame. Thereafter, the trained Faster R-CNN model (mentioned in Section 5) was employed to detect vehicles frame by frame, where the obtained features were utilized to update the corresponding parameters in the Kalman filter equations for the sequential frames. Meanwhile, the obtained state variables and covariance errors were utilized as the input information to predict the position of the target object in the next frame, and the data association algorithm was employed to establish the mapping relationship between the built target tracking set with the predicted target detection set, which can achieve real-time updating of the tracking results. The step was continuously cycled until the tracking vehicle disappeared out of sight. The flowchart of the vehicle tracking process is shown in Fig. 15.

A field experiment was conducted to validate the reliability of the tracking method. The Guanyingang Bridge, located on the Xiangjiang River in Changsha, China, was selected for traffic tracking research, and a Canon 6D camera installed on 1.6 m tripod was arranged at the western bridge approach. The resolution and recording frame rate of the monitoring video was defined as

![Example of vehicle tracking results.](image)

Fig. 16. Example of vehicle tracking results.

![GUI operating interface of the vehicle information identification system.](image)

Fig. 17. GUI operating interface of the vehicle information identification system. (Note: the red dotted window denotes the control options for corresponding vehicle information identification modulus, and the red solid line denotes the distinguished trajectory route of the vehicle. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
720 × 560 pixels at 25 fps respectively in order to enhance the computation efficiency. To effectively evaluate the proposed algorithm, a 57 s clip was extracted from the monitoring video, which included the motion of a light-coloured mini bus and a dark-coloured 2-axle truck with 6 wheels. The clipped video was transformed into 1440 continuous frame images for calculation purposes. In the test, the corresponding parameters of the Kalman filter was initialized according to the suggestions of Bolme et al. [57]. A threshold control coefficient was further introduced to determine whether the target object existed in the monitoring region.

The calculation process was conducted on the Matlab software platform with the tracking results of the target vehicles are shown in Fig. 16. It can be seen that the vehicles have been clearly captured in the continuous frames, indicating that the motion state of a target vehicle can be effectively tracked in real life. Although the colour of the 2-axle truck with 6 wheels is similar to the background, which could have caused confusion during the tracking process, the robust tracking ability of the proposed method was nevertheless able to distinguish the vehicle.

7. Automatic monitoring system

This study has developed a vehicle information identification system (known as the Automatic Vehicle Loads Identification System of HNU) based on a Matlab graphical user interface (GUI) platform along with the algorithms explained earlier in this paper. The system includes four modules: the vehicle weight prediction module, the vehicle types classification module, the vehicle object detection module, and the target vehicle tracking module.

The GUI operation platform is shown in Fig. 17, where the information details of the recorded video can be referred to at the foot of the interface. The progress bar is arranged below the display window so that users can manually shift it to observe identification results in any of the frames. Control options for vehicle information identification in weight prediction, classification, detection and tracking modulus are presented on the right side of the interface. When a vehicle moves into the virtual monitoring region, the video recorded by the roadside surveillance camera is first transformed into frame sequences for computation, and then the trained VGG-16 and Faster R-CNN models can be loaded to conduct the vehicle types classification and object detection in each frame, respectively. Based on the statistical results of vehicle parameters, the weight information of a classified target can be predicted directly from the monitoring videos. Taking the mini bus as an example, it was found that both the corresponding type and position information could be effectively distinguished from a captured frame. The proposed method also allows for the weight of a classified vehicle to be predicted by non-contact machine vision technology instead of relying on a costly BWIM system. The main advantage of recognizing the weight interval is that it is beneficial to automated structural identification methods based on big data [58]. Furthermore, the calculated trajectory line is consistent along with the motion direction of a vehicle, which could play a significantly role in bridge performance evaluation.

8. Conclusions

This study proposes an automatic non-contact methodology to distinguish vehicle loads on bridges based on deep learning algorithms and machine vision technology. Vehicle information is obtained in the monitoring video recorded by a roadside traffic surveillance camera. The automatic identification method consists of four modules: the vehicle weight prediction module, the vehicle types classification module, the vehicle object detection module, and the target vehicle tracking module. A field test was conducted to evaluate the reliability of the system. The main conclusions are as follows.

(1) Rough-grained classification has been conducted on common vehicles, which were classified into nine different types. The distribution interval for axle weight and spacing of each type were obtained from the statistical results of 8402 messages, through which the mapping relationship between vehicles and their corresponding weight information was built.

(2) The DCNN was trained by an established dataset of 8624 vehicle images, with nine vehicle types used to enhance the generalizability of the network. By combining transfer learning strategy with the generic features extracted from ImageNet, the test accuracy of AlexNet was enhanced from 85.37% to 93.46%. The highest accuracy of 98.17% was achieved by VGG-16 under the strengthening learning strategy, indicating that vehicle types can be effectively classified by monitoring videos.

(3) The Faster R-CNN with the co-shared VGG-16 model was employed to detect a vehicle’s position in video frames. The vehicle images with various test conditions (i.e., types, target dimensions, and lighting conditions, etc.) were utilized to evaluate the performance of the trained Faster R-CNN. The results indicate that a target vehicle could be effectively distinguished with accurate identification and position information, which attests to the robustness of the monitoring video.

(4) Real-time tracking research for motion vehicles was conducted by the Kalman filter, where the corresponding parameters in each frame were updated with results from the Faster R-CNN. The field test results demonstrated that the vehicle motion state could be effectively tracked in the monitoring video, even if the vehicle colour is similar to that of the background.

Declaration of Competing Interest

The authors declared that they have no conflicts of interest to this manuscript. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the manuscript submitted.

Acknowledgements

The authors sincerely appreciate the funding support provided by the National Natural Science Foundation of China (No.51878264), the National Key Research and Development Program of China (No. 2016YFC0701400, 2016YFC0701308), and the Key Research and Development Program of Changsha City (kq1801010).

References


