A Vision-based Overload Detection System for Land Transportation

Yuntao Wang1*, Chengxi Xia2*, Haibo Sun2, Yihan Zhang3, Zheyan Liu3, Yufei Wang4, Naixuan Xu4, Jianjia Zhu4 Yuchen Zhang4, Huaqiang Wu2, Yuanchun Shi12

1Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
2Global Innovation Exchange (GIX) Institute, Tsinghua University, Beijing, 100084, China
3Beijing University of Posts and Telecommunications, Beijing, 100876, China
4Trucker (Beijing) Technology Co. Ltd, Beijing, 100028, China
5Institute of Microelectronics, Tsinghua University, Beijing, 100084, China
Email: yuntaowang@tsinghua.edu.cn, {clytze20, sxndqc}@gmail.com, {wangyufei, xunaixuan, zhujianjia, zhangyuchen}@kachexiongdi.com, {wuhq, shiyi}@tsinghua.edu.cn

ABSTRACT

Overloaded trucks pose a severe threat to highway traffic, increasing the rate and severity of accidents, while damaging road infrastructure. Enabling an efficient and low-cost overload detection method at existing weighing stations would facilitate regulatory compliance. This paper presents a real-time, accurate truck overload detection system that leverages existing surveillance cameras installed at weighing stations. To achieve this goal, we applied computer vision algorithms on video from an indoor camera monitoring a digital display and on another outdoor camera monitoring the station’s weighing bridge. The truck’s actual weight is obtained by reading the numeric digit display on images from the indoor camera, while the truck’s maximum load capacity is estimated by recognizing the its wheel layout using the video from the outdoor camera. Through evaluation using video data from a weighing station, the optical digit recognition algorithm achieves an accuracy of 99.0%, while the load capacity estimation algorithm achieves an accuracy of 93.18%. Furthermore, our system can achieve an accuracy of over 90% among 21 overload events. Finally, we conclude with a discussion of potential optimizations and other future work.

Keywords: Overload detection, numeric digit recognition, vehicle classification, smart weighing station

INTRODUCTION

Overloaded trucks pose a severe threat to transportation operations, increasing risks for drivers and negatively impacting road infrastructure as well as best practices among different transport modes and operators (Turner, 2013; Jessup, 1996). Currently, overload detection depends mainly on traffic police at roadside weighing stations, which is inefficient and causes traffic jams. A more advanced system such as a weigh-in-motion (WIM) overload detection device (Deesomsuk, 2009, Jacob, 2010) is costly. It also requires renovating existing infrastructure, which makes it hard to

* The two authors contributed equally to this work.
deploy. As a result, there are a limited amount of overload control centers. With a low probability of being caught, drivers overload their vehicles to pursue higher profits.

In China, a weighing bridge is a standard device installed in weighing stations or factories that manufacture bulk commodities, such as coal and steel. It can measure the weight of the goods or the weight of each truck. The owners of the factory usually install surveillance IP cameras in the weighing room for safety and monitoring purposes, as Figure 2 shows. Enabling low-cost overload detection methods on these existing weighing stations would transfer them into overload detection infrastructures, which can significantly promote regulatory compliance.

For the first time, we propose a low-cost computer vision-based overload detection system that can be easily deployed in any weighing station. We achieved this by applying real-time computer vision algorithms on videos from existing surveillance cameras installed at weighing stations. First, we developed an algorithm to recognize the indoor digital display, which shows the truck’s actual load using the indoor camera. Then we utilized the video from the outdoor camera to estimate a truck’s maximum load capacity by recognizing the layout of the truck’s tires and mapping the truck axle features to maximum load capacity. Using these two algorithms in conjunction, we developed a real-time and accurate overload detection system. Our contributions in this paper are as follows:

1) We propose an optical digit recognition algorithm based on one-dimensional mapping on blurred images with 99% accuracy to detect the actual truck weight.
2) We propose a maximum load capacity recognition algorithm that identifies the truck’s axle layout from images with a dramatically inclined view. The algorithm achieves 93.2% axle recognition accuracy.
3) We build a real-time, low-cost, and easy-to-deploy overload detection system leveraging existing surveillance cameras at weighing stations, which achieves 90% overload recognition accuracy.

RELATED WORK

In this section, we describe existing overload detection technologies as well as the difference between our work and previous work. To our best knowledge, no related work has explored computer-vision based approaches relevant to our work.

Currently, weighing bridges and wheel-and-axle scales are the most frequently used system to measure gross vehicle weight. However, this approach relies heavily on human labor at the weighing station, which is inefficient and often causes traffic jams. To improve the efficiency of weighing operations, a variety of advanced weighing systems have been developed. A weigh-in-motion (WIM) system is installed in concrete or asphalt platforms with a minimum length of 30 to 40 meters. Embedded weight scale sensors can accurately measure the actual load of the truck, while optical sensors measure the maximum truck load based on truck’s axel structure. This system works at speeds from 5 to 15 km/h (Deesomsuk 2009). Other systems can support high-speed WIM by measuring the axle and vehicle’s actual loads when vehicles are moving at a certain range of speed. This is a fully automated weighing system, which can detect whether an individual vehicle is overloaded. More advanced WIM systems work in conjunction with cameras to get more data about the truck, like the plate number (Jacob, 2010). Some researchers explored solutions enabling weighing bridges to detect vehicle overload by adding more sensors (Jiang, 2013; Hu, 2011; Qiao, 2018). However, these overload detection systems are costly and require infrastructure/equipment modification, which is not easy to deploy or scale.
Recently, to make overload detection more accessible, researchers have explored computer-vision-based approaches. Some researches tried to measure the distance gap between the vehicle and the ground (Zhou, 2019) or vibration pattern (Ding, 2019) to estimate the load weight using a vision-based algorithm. However, they only evaluated certain car models and the algorithm relies heavily on the calibration of each camera for a given vehicle model. Researchers have also applied convolutional neural networks (CNN) to distinguish the overload and non-overload trucks through image recognition (Zhou, 2018). However, this work failed to provide details about the ground truth labeling method and a reproducible dataset along with the machine learning method. Furthermore, this system was unable to provide reliable overload behavior evidence or achieve high accuracy.

In contrast with previous work, our technique deploys overload detection capability to any existing weighing stations by applying computer vision algorithms on the existing surveillance system. This software-based solution is low-cost, effective, and easy-to-deploy.

**SYSTEM OVERVIEW**

As shown in Figure 1, the Overload Detection System (ODS) contains three components that are the maximum load capacity detector (MWD), the actual weight detector (AWD), and the truck pass detector (PD). The maximum load capacity detector is used to detect the permitted load limit of the truck on the weighing bridge, and the actual weight detector is used to acquire the loaded weight of the target truck. By comparing these two results, the system can infer if the truck is overloaded or not. The pass detector is designed to detect the existence of a vehicle as well as its inbound or outbound event.

**DATA ACQUISITION AND APPARATUS**

We identified one coal manufacturer with an in-factory weighing station in Jiexiu, Shanxi Province, China. The original purpose of this weighing station was to obtain the weight of the loaded freight on the truck, as a means to determine pricing for its transportation. Recent local regulations of
truck-based logistics required that surveillance cameras be installed in the weighing station with recorded videos can serve as supporting evidence for safety, accidence or law enforcement monitoring. We utilized two cameras with one in the weighing room and another facing the weighing bridge as Figure 2 shows. The dataset we used in this paper is collected from these two surveillance cameras. We obtained 10 days of historical video data on which we labeled 912 valid frames containing the whole numeric digit display. We also labeled 426 valid outbound pass sequences including 21 overload behaviors into three-axle layout classes. There are 19/126/281 samples for the 4/5/6-axle classes.

We conducted the training process on a Nvidia® TITAN XpTM GPU server with 12GB memory and tested the system on a 2.8 GHz Intel® CoreTM i7 processor with 16 GB memory with 64-bit OSX operating system.

![Figure 2 Sample images from indoor camera (A) and outdoor camera (B)](image)

**ACTUAL WEIGHT DETECTION BY RECOGNIZING DIGITAL DISPLAY**

The digit recognition algorithm recognizes the actual weight of the truck by interpreting the numbers shown on the digital display during the weighing process. Furthermore, this algorithm can also be used to sense the onboard process of an outbound truck, as the input of the vehicle pass detector.

As shown in Figure 2A and Figure 3A, because of the long distance between the ceiling-mounted camera and the digital display, the numeric digits are blurred and close together. This proves to be too much of a challenge for typical number recognition or character recognition algorithms (Saghaei, 2016, Netzer, 2011). Therefore, we proposed a one-dimensional mapping recognition method on the segmented and perspective transferred low-resolution numeric digit display images.

![Figure 3. An original picture of the digital display (A), a demonstration of color-based clustering (B), and the segmentation result of the display (C).](image)
**Digit Display Identification.** Our system should identify the numeric digit display quickly and robustly while minimizing the need for customization. To reduce computational requirements, we first manually marked out an interest region on the video where the screen is placed for a particular station. We localized the contour of the display by applying color-based clustering (Jian, 2007) at the pixel level.

**Numeric Digit Recognition.** After localizing the digital display, we applied perspective transformation for better recognition performance. We utilized a projective mapping method projecting the image onto a different view plane (Mezirow J, 1991). The numeric digits recognition has the following steps to identify each digit’s location and number.

![Figure 4](image.png)

*Figure 4. The two display modes on display after perspective transformation and scaling (A) and the one-dimensional features extracted from two-dimensional images (B).*

**Step 1. Templates building.** We manually selected and blurred templates of digit 0 to 9 from the dataset as the templates for template matching (Jurie, 2001).

**Step 2. Template matching.** We conducted a 2D template matching on the region of interest (Figure 4A) with all templates. The equation shown below demonstrates the process and results.

\[ R_{ek} (x, y) = [x_0, x_1, \ldots, x_l] \]

\( R_{ek} \) refers to the template matching function on the k-th template and \( Res_k \) indicates the similarity result between the region begins at \((x, y)\) with the template. \( x_i \) refers to the result on column \( i \) which is an array of matching ratio and \( l \) denotes the length of \( Res \).

**Step 3. Max pooling.** We conducted max pooling on \( Res_k \) among each column to reduce the feature quantity as the digits are queued horizontally. So for each digit template k, the result can be described as \( S_k = [\max(x_0), \max(x_1), \ldots, \max(x_l)] \), Where if elements in \( S_k \) is above a threshold, it is regarded as a possible digit position, \( S_{k,\text{filtered}} \).

**Step 4. Digit localization.** We conducted a greedy algorithm with some boundary conditions to find the maximum possible digits and positions on the display. The boundary conditions are 1. the last (rightmost) four digits would always be presented in both display modes (Figure 3C), with the last digit to be 0 as the sensitivity of the scale on the weighing bridge is 20kg. 2. The gaps between nearby digits are equal. We checked the matching ratio result of digit 0’s template reversely and to find the possible digits at those positions with the standard gap distance. The following equation shows the algorithm:

\[ p1 = p \pm x \pm \sigma, p2 = p \pm 2x \pm \sigma, p3 = p \pm 3x \pm \sigma, p4 = p \pm 4x \pm \sigma \]
Where \( p_1, p_2, p_3, p_4 \) are the possible positions for the last four digits which should be in the \( S_{k, \text{filtered}} \), and \( x \) is the distance between each digit. \( \sigma \) represents the system error, and \( p \) refers to the initial position and traverse from the first possible position to the last position. The goal is to find the largest \( p \) to fit the equation with the last digit as 0 with the highest confidence. The corresponding templates of the peak values of the matching ratio at \( p_1, p_2, p_3, p_4 \) are the inference digit for that position respectively.

**Step 5. Display modes check.** When the last four digits are settled, we checked the possible position for the \( p_0 \) (5-figure display mode). If the maximum matching ratio of templates (1-9) among that position larger than the threshold, it displays in the 5-digit mode. Similar to the rest four digits inference, the corresponding template is the inference result.

![Diagram](image)

**Figure 5. The overview of the proposed digit recognition algorithm**

**Evaluation and Result.** We only discuss the recognition accuracy using valid frames that contain the whole numeric digit display. The recognition accuracy is 99% as shown in Table 1. This algorithm can process 18 frames per second on our testing apparatus, which makes it possible to run in real-time without significant delay.

<table>
<thead>
<tr>
<th>Number of frames</th>
<th>Valid</th>
<th>Correctly recognized</th>
<th>Unrecognized</th>
</tr>
</thead>
<tbody>
<tr>
<td>912</td>
<td>903</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Ratio (%)</td>
<td>100</td>
<td>0.99</td>
<td>0.01</td>
</tr>
</tbody>
</table>

**Table 1. The performance of digits recognition**

**VEHICLE PASS DETECTION**

We define the vehicle pass detection as the temporal function \( F_{pass}(t) \) (\( F_{pass} = \{0, 1\} \)). If the vehicle has not passed yet or has already passed through the Region of Interest, we define \( F_{pass} = 0 \). If the vehicle is passing, \( F_{pass} = 1 \). The passing detector is to estimate \( F_{pass} \). We implemented the vehicle pass detector with the following steps:

**Step 1. Data filter.** We applied a one-second moving window to screen a potential pass event on the actual weight data stream recognized from the numeric digit display. Once the value exceeds our maximum difference threshold of 500kg in the one-second, step 2 is triggered.

**Step 2. Direction check.** We utilized the video data from the outdoor camera to detect whether the direction of the truck is inbound or outbound. As the camera sits focused on a fixed
view, the pixel changes are dramatically different when a vehicle is coming from a different direction, which enables us to infer the direction of its movement.

**Step 3. Valid pass check.** We added some boundary conditions to verify a valid pass event (i.e. to filter out noise in the data caused by other vehicles). The vehicle takes about 20 seconds to fully accomplish the inbound and stays onboard about 1 minute. The truck’s weight should be at least 10,000 kg when it stops on the weight station. Meanwhile, the resolution of the weighing bridge is 20kg, which makes the detector more tolerant to noise.

Signal fusion is applied to provide a robust function with direction information. The pass detector could be simplified as the following formula.

\[
\hat{p}_{\text{pass}}(t) = \begin{cases} 0, & \Delta w < \epsilon \\ 1, & \text{else} \end{cases}
\]

\( w \) represents the actual weight acquired from the weighing room’s camera, \( \Delta w \) represents the differential value of \( w \) between frames, and \( \epsilon \) represents the tolerance of the system error.

**Evaluation and Results.** The results in Table 2 indicate that our method can detect vehicle pass with an accuracy of nearly 100%. The only missed sample was caused by a vehicle stopping in the middle of the onboarding process, which the system mistakenly identified as the end of the process.

<table>
<thead>
<tr>
<th>Number of passes</th>
<th>Valid passes</th>
<th>Recognized</th>
<th>Missing</th>
<th>Unrecognized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio (%)</td>
<td>426</td>
<td>425</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>99.998</td>
<td>0.002</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2. The performance of vehicle pass detector

MAXIMUM LOAD DETECTION BY RECOGNIZING LAYOUT OF TRUCK AXLES

This algorithm to acquire the maximum load capacity using the video from the outdoor camera monitoring the weighing bridge is shown in Figure 2B. According to the *freight vehicle overload identification standard*, each unique axle arrangement corresponds with a specific maximum load capability. Therefore, the maximum load capacity can be estimated by recognizing the truck’s axle layout.

Since the outdoor camera was initially designed to record the truck’s plate number and the weighing bridge, we are unable to obtain a clear view of the truck’s side. Therefore, a classic axle detection method with perspective correction (Grigoryev, 2015) is not viable. The major technical challenge is recognizing axle layout using the images with an inclined angle. However, we observed that the image region near the camera contains enough visual information about the wheels to determine the axel layout. Based on this finding, we proposed a method that takes a panoramic photo utilizing the video sequence, which includes the whole truck onboarding procedure as Figure 6 shows.

**Step 1. Region of interest selection.** We selected an image region near the camera with a clear view of the truck wheel as our ROI, and for each image in the video sequence of an outbound onboard procedure, we cropped out that region. This step can be represented as \( ROI_i = S_i^T \ast F \). \( S_i \) refers to the source frame with timestamp \( i \), \( F = [0, 0, 0...1, 1...1, 0, 0...0]^T \) is a region selection vector.

† [http://www.gov.cn/xinwen/2017-11/26/5242386/files/30a262fd236f426cbd5e6f9c34f1c43e.doc](http://www.gov.cn/xinwen/2017-11/26/5242386/files/30a262fd236f426cbd5e6f9c34f1c43e.doc)
**Step 2. Panoramic photo construction.** A fusion is conducted after each ROI been cropped to form \( D = [ROI_0, ROI_1, ..., ROI_T, ...] \) with all the truck wheel information for axle arrangement estimation. Compared to Figure 2B, the panoramic photo shows clearer axle arrangement.

**Step 3. Photo slicing.** We sliced the photo into uniformed sub-images which contains few continuous columns of the photo. This step servers two purposes. Firstly, as the panoramic photo has various widths that depends on the onboard time of each truck, this step helps us to uniform the data point. Furthermore, this method expanded the data size by 700 to 1200 times using collected hundreds of panoramic photos.

![Figure 6. The feature split algorithm running on the panoramic photo of the truck](image)

**Step 4. Binary classification on the wheel.** There are two potential statuses for all the sliced photos, they either contain a part of a wheel or they don’t. For the binary classification, we applied and tuned ensemble learning to get the best result. We compared methods including AdaBoost, Extra Trees, Random Forest and XGBoost. Figure 7 and Table 3 illustrate that the Extra Trees method reaches the highest accuracy with the fewest estimators. Therefore, we deployed the Extra Trees method in the ensemble-based algorithm.

To improve performance in a real-world deployment, instead of conducting step 2, step 3 and step 4 sequentially, we could conduct them in pipeline, which means that we do binary classification for each image in the valid outbound pass video, after getting the ROI, skipping step 2 and step 3, which saves space and time.

<table>
<thead>
<tr>
<th>Number of Estimators</th>
<th>AdaBoost</th>
<th>Extra Trees</th>
<th>Random Forest</th>
<th>XGBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>0.8768</td>
<td>0.9970</td>
<td>0.9969</td>
<td>0.9563</td>
</tr>
</tbody>
</table>

**Table 3. Binary classification results for different ensemble learning methods**

![Figure 7. Finetuning results using different ensemble learning methods](image)
**Step 5. Axle arrangement estimation.** Based on the binary classification results, the algorithm can recognize the axle layout based on two rules: First, the wheel region should contain highly intense valid results with little or no gap smaller than a predetermined threshold. Second, the gap between different wheel regions should be distinguishable and larger than our predetermined threshold.

The axle layout classification results are shown in Table 4 with an average accuracy of 93.19%. Most of the errors happen with images taken at night because of the poor lighting conditions. Classification results on the daytime data increase by 97.68%. On our testing apparatus, the whole process for a single pass takes 1.8 seconds on average.

<table>
<thead>
<tr>
<th>Type</th>
<th>4-axle</th>
<th>5-axle</th>
<th>6-axle</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sequences</td>
<td>19</td>
<td>126</td>
<td>281</td>
<td>426</td>
</tr>
<tr>
<td>Correctly recognized</td>
<td>18</td>
<td>113</td>
<td>266</td>
<td>397</td>
</tr>
<tr>
<td>Accuracy (%)</td>
<td>94.74</td>
<td>89.68</td>
<td>94.66</td>
<td>93.19</td>
</tr>
</tbody>
</table>

Table 4. Truck axle layout classification result

<table>
<thead>
<tr>
<th>Type</th>
<th>4-axle</th>
<th>5-axle</th>
<th>6-axle</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sequences</td>
<td>12</td>
<td>83</td>
<td>207</td>
<td>302</td>
</tr>
<tr>
<td>Correctly recognized</td>
<td>12</td>
<td>80</td>
<td>203</td>
<td>295</td>
</tr>
<tr>
<td>Accuracy ratio (%)</td>
<td>100</td>
<td>96.38</td>
<td>98.07</td>
<td>97.68</td>
</tr>
</tbody>
</table>

Table 5. Daytime truck axle layout classification result

Finally, we combine all of the above components to finalize our Overload Detection System (ODS). The system successfully detected 19 overloaded behaviors among 21 overloading cases within the 426 valid pass sequences with one false recognition and one failed recognition.

**DISCUSSION AND FUTURE WORK**

This paper presents an automated, real-time and accurate truck overload detection system. This is achieved by applying computer vision techniques to videos from the existing surveillance cameras at weighing stations. We described the core methods and algorithms to realize this system. We proved our system’s feasibility and effectiveness by evaluating our algorithms’ performance on real-life video data. We discuss some limitations and future work in this section.

Our system is based on the observation that numerous weighing stations have surveillance camera systems. However, we only tested our system’s feasibility and performance using data from one station. Our next step is to deploy this system to other weighing stations and optimize our methods with more training data.

Although our computer-vision-based method can be easily deployed on existing weighing stations, it suffers from limitations due to occlusion and lighting conditions. While our system detects the truck’s wheel layout to estimate its maximum load capacity, the wheel layout and the maximum load capacity are not always characterized by a one-to-one-mapping relationship. The system can remind the operators in the weighing room to confirm the vehicle’s parameters for more precise detection.

The algorithm proposed and developed in this paper requires limited computing resources. Therefore, our system can be deployed on an edge computing module or a chip in the surveillance camera system for large scale deployment. Furthermore, reinforcement learning could be applied
in the future with human assistance for correcting the inaccurate inference of overload detection to learn from data in daily use and improve the system’s long-term performance.

ACKNOWLEDGEMENT

This work is supported by the Undergraduate Education Innovation Grants and the Graduate Education Innovation Grants, Tsinghua University. Our work is also supported by the Beijing Key Lab of Networked Multimedia and CERNET Innovation Project (NGII20170324), CERNET Innovation Project (NGII20170801) and Trucker (Beijing) Technology Co. Ltd.

CONCLUSION

This paper presents an automated, real-time, and accurate truck overload detection system that leverages existing surveillance cameras at weighing stations. The truck’s actual weight is recognized by reading the digital display using the video from the indoor camera with an accuracy of 99.0%. The maximum load capacity is obtained by recognizing the axle layout using images from the outdoor camera with an accuracy of 93.18%. Finally, our system can recognize overload behaviors with an accuracy of over 90%. The high recognition accuracy and no external hardware requirements demonstrate our solution to be a feasible, effective, and easy-to-deploy overload detection system.

REFERENCES


Qiao, Xin-Xin, and Yi-Ding Zhao. (2018) "Vehicle overload detection system based on magnetoresistance sensor," 2018 International Conference on Electronics Technology (ICET), Chengdu, 2018, pp. 102-105. DOI: 10.1109/ELTECH.2018.8401434

