

An Intelligent IoT-based Vision System for Nighttime Vehicle Detection and Energy Saving

Chih-Hsien Hsia,^{1*} Shih-Chieh Yen,² and Jing-Han Jang¹

¹Department of Computer Science and Information Engineering,
National Ilan University, Taiwan

²Department of Electrical Engineering, Chinese Culture University, Taiwan

(Received February 28, 2019; accepted May 7, 2019)

Keywords: energy saving, vehicle recognition, vehicle light detection, vehicle counting, nighttime traffic

In recent years, the rise of industrial societies and the Internet of Things (IoT) has encouraged the growth of vehicular transportation, which, in turn, has led to intelligent transportation systems (ITSs) becoming an important field of research. In view of this, nighttime vehicle detection and the counting technique that will facilitate future research on energy saving are presented. In this work, the Hough transform was first performed to detect lane lines in an image, and subsequently, all images underwent feature enhancement processing. Next, the light source data from the detected line of each lane was obtained, and at the same time, the aspect ratio and spacing of light source pairs were computed in order to determine if they matched the set values and to perform vehicle counting. Furthermore, since driving habits differ and some vehicles would straddle the lane line, an approach to recognizing and counting lane-straddling vehicles in order to avoid misjudgments is proposed in this study. Lastly, LED lights were used to simulate street lights and controlled on the basis of the traffic volume data obtained in the manner described earlier. The experimental results show that the proposed technique could be effectively utilized to perform nighttime vehicle detection and counting since it achieved a high average correction rate of 94%, as well as a computing time of 44 frames per second.

1. Introduction

To allow for the continued expansion of the transportation capacity of an urban transport system, it is necessary to integrate sensor technologies into the existing infrastructure in order to enhance intelligent detection and monitoring and achieve a higher service efficiency. The resulting system is referred to as an intelligent transportation system (ITS). In highly automated countries, ITSs play a crucial role in the key trends that will guide the long-term development of the world's transportation systems. An ITS comprises multiple subsystems, including advanced traffic management systems (ATMSs), advanced public transportation systems (APTSs), and

*Corresponding author: e-mail: chhsia625@gmail.com
<https://doi.org/10.18494/SAM.2019.2351>

advanced vehicle control systems (AVCSs). However, an ATMS primarily collects various types of data concerning traffic flow. In recent years, countries across the world have experienced an increase in vehicle usage, which has exacerbated traffic congestion and prompted the implementation of manual traffic monitoring systems as a solution to the problem. The advancement of computer vision technology has enabled the widespread application of image-based sensor detection methods in intelligent traffic monitoring. In addition, computer vision technology outperforms other detection methods in terms of the cost and ease of installation. Furthermore, energy conservation has also become an important issue in recent years. If street lights are operated at a higher power level and maintained at a higher brightness level during off-peak nighttime periods when the traffic volume is low, excessive power consumption will occur and lead to resource wastage during these periods. Hence, in this work, we aimed to enable power control for street lights through traffic volume monitoring, such that an intelligent power (brightness) control for street lights can be achieved.

Yeh classified existing vehicle detection sensors into three categories, namely, active, passive, and hybrid sensors.⁽¹⁾ The detection method proposed in this study utilizes passive sensors and is a vision-based vehicle detection system. Image preprocessing is performed using the method proposed in a previous study and the image capture device is positioned at the center, as was proposed in other studies.^(2,3) Yuan *et al.* proposed the combined use of Otsu's method and the top-hat method to obtain the optimal binarized image, as well as the optimal data during binarization.⁽⁴⁾ Even though they utilized a divided lane approach that enabled detection even with low-intensity light sources, the said approach also led to noise data being misjudged as vehicle lights and to excessively long computation times. Zhang *et al.* proposed an edge detection approach that utilized a Laplace of Gaussian (LoG) filter instead of the region-of-interest (ROI) approach.⁽⁵⁾ However, the proposed method led to unnecessary computation and longer computation times during the postprocessing for vehicle light detection. Robert proposed a vehicle detection method that considered the eigenvalues of windshields. Although the detection rate of this method was relatively more stable during the day as the daylight environment was sufficiently bright, the lack of ambient light sources at night made it difficult to detect vehicle windshields and resulted in misjudgments.⁽¹⁰⁾ Jurić and Lončarić⁽¹¹⁾ adopted the method proposed in Ref. 12, which did not require manual parameter adjustments and boasted simpler processes that reduced the amount of computation needed, but the results of that study indicated a low vehicle correction rate. In Ref. 13, the use of Haar-like features was proposed as a method of increasing the computational speed. However, this approach was prone to misjudgments when vehicles are parallel or when image occlusion occurs. Chen *et al.* proposed the application of the Nakagami distribution to filtering to obtain an appropriate ROI and used deformable part models (DPMs) to eliminate unwanted background information and detect vehicles.⁽¹⁴⁾ However, this method also filtered out details of data that were still relatively useful. Park and Song proposed the dark channel prior (DCP) method, which is often used to perform image defogging and imposes a relatively heavy computational load.⁽¹⁵⁾ Chen *et al.* proposed the use of support vector machines (SVMs) to address classification problems, and while this method allowed for a higher correction accuracy, it also required a large memory space.⁽¹⁶⁾ Then, various types of vehicle technology based on artificial intelligence have been published, such as deep learning and machine learning.^(17,18,20)

To overcome the classification problem, improving the detection methods that we previously published is aimed in this study.⁽⁶⁻⁹⁾ The previous techniques that these methods applied involved the use of two detection lines, and after deleting duplicate vehicle counts, deletion errors would sometimes occur when two vehicles moving in parallel crossed a line at the same time. To address this issue, a separated-lane counting method that addresses the errors related to the duplicate counting problem and uses a single detected line to count and delete duplicate counts and, thereby, allows for a higher computational speed is proposed in this work. The lane separation method is based on the methods used in other studies and modified by incorporating Canny edge detection and the Hough line transform to determine lane lines.^(8,9) Lastly, the system also utilizes real-time vision information technology to detect nighttime vehicles and simulates street lights using LED lights to implement power control. When linked to LED street lights, the system will enable sending traffic volume data collected at one road intersection to the street lights at the next intersection of LED street lights using the Internet of Things (IoT), such that their brightness can be controlled in a simulated manner to enable energy conservation.

The rest of this paper is organized as follows. In Sect. 2, research objectives and basic principles are briefly introduced. The proposed nighttime vehicle detection approach and efficient system architecture are presented in Sect. 3. Section 4 shows performance comparisons with other existing nighttime vehicle detection systems. Conclusions are given in Sect. 5.

2. Research Objectives and Preprocessing

In general, advanced cities will eventually have to confront the problem of traffic congestion, which could be alleviated through a system that accurately delivers real-time traffic data.^(19,20) As vehicle detection technology is approaching maturity, various methods are now available and are distinguished primarily by cost. At present, nighttime vehicle detection technology is relatively less mature than its daytime counterpart, since nighttime factors such as the lack of light and road reflections on rainy days have created detection issues that increase the difficulty of nighttime detection.^(6,7)

During the off-peak nighttime period (i.e., midnight), there are practically no vehicles on the road, and operating street lights at an excessively high power level during this time would result in energy wastage, as shown in Fig. 1. In this work, an algorithm for performing the separated-lane counting of nighttime vehicles is proposed, and LED lights, which will fully replace current street lights in the future, are used to simulate street lights (the power and brightness of the LED lights are adjustable via an intelligent control system). The algorithm allows for the nighttime traffic volume to be monitored, as well as the use of such data as a basis for adjusting the power level of street lights at the next intersection, with the aim of reducing electric energy wastage. Thus, in this study, we explored ways in which images could be stored and returned, how computer vision technology could be incorporated to enable real-time nighttime vehicle detection, and how the brightness of street lights could be controlled.



Fig. 1. (Color online) Nighttime traffic during the off-peak period.⁽⁷⁾

2.1 Image binarization and its denoising

An image was first read and converted to a grayscale format, and the preprocessed ROI was then converted to a binarized image. A comparison of pixel grayscale values and the preset threshold (T) was then performed according to Eq. (1) shown below.

$$b(i, j) = \begin{cases} 1, & f(i, j) \geq T \\ 0, & f(x, y) < T \end{cases} \quad (1)$$

Next, the binarized image $b(i, j)$ values were input, and connected components were used to filter out blocks with areas that were smaller than the set value. The area sizes of the connected components were then computed, and deletions were performed after determining if the area values were smaller than the set value. When the area of a light source was smaller than the set value P , a deletion would be performed in order to highlight the vehicle light.

$$N8(p) = N4 \cup \begin{matrix} (i+1, j+1), (i+1, j-1) \\ (i-1, j+1), (i-1, j-1) \end{matrix} \quad (2)$$

$$b(i, j) = \begin{cases} 1, & \text{otherwise} \\ 0, & N \leq T \end{cases}$$

2.2 Rectangularization of vehicle lights

Primarily, this method was performed for the connected component detection and labeling of binarized images. Binarized image data were used to label surrounding pixels and determine

if they were part of the same component. In the work, we utilized eight connected components, with the eight marking points being the top left, top, top right, left, right, bottom left, bottom, and bottom right points. After marking the eight points, they were each given different weights that were added together to determine if the middle pixel should be 1 (white point) or 0 (black point) and, thus, if it was part of the same component. Eight connected components were used to account for the data neighboring the pixels to be processed, and to convert irregularly shaped vehicle lights to a more squarelike shape.⁽⁷⁾ Since the remaining white blocks (vehicle lights) were irregularly shaped, and in order to increase the system accuracy and reduce misjudgments, each white block was converted to a rectangle shape to facilitate the computation of the width and height of each vehicle light, and to subsequently confirm that said light was a vehicle light.

Dilation is an application of the morphology used in image processing, and its results are computed using input structuring element (SE) values. The dilation of image A by SE B is defined by $A \oplus B$, and its output results exhibit an outward expansion effect.

3. Nighttime Vehicle Detection System

Figure 2 shows the flowchart proposed in this work, which was primarily divided into initialization, preprocessing, vehicle light detection, vehicle recognition, vehicle counting, and Arduino platform control phases.

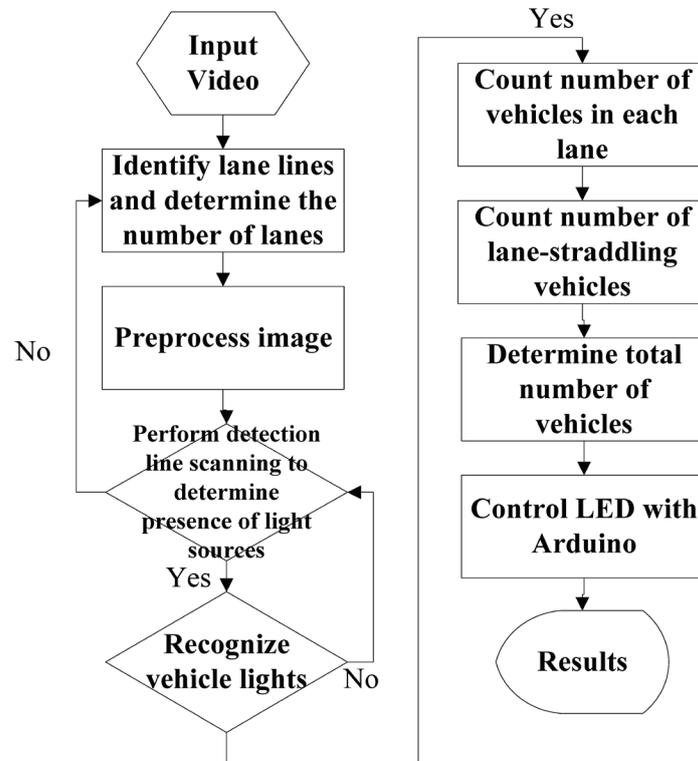


Fig. 2. Proposed flowchart for nighttime vehicle detection.

3.1 Identification of lane lines

The first image to be read was converted to a grayscale image, the average grayscale values of the ROI were then determined, and the values between $1.3\times$ and $2.95\times$ were converted to 1 (white point), while the rest were converted to 0 (black point). In fact, we have quantized the range between 0 and 1, and tested for all of the possibilities for ROI's average grayscale values, and according to the experimental result, when using $1.3\times$ or $2.95\times$, the best result can be achieved. Thereafter, the Hough transform was performed on the image, so as to execute line detection and identify the lane lines (see Fig. 3). To increase the efficiency of line detection, the detection angle range was set to be larger than 130 degrees and smaller than 70 degrees, and thereafter, the detected lane line positions were transmitted to the main program.

3.2 Image preprocessing

After the image was input, the following preprocessing steps were executed:

Step 1 (Grayscale processing of image): The read image would undergo grayscale processing, which converted a color image to a grayscale image to facilitate subsequent steps, as shown in Fig. 4(a).

$$\text{Gray scale} = R \times 0.299 + G \times 0.587 + B \times 0.114 \quad (3)$$

Step 2 (Setting of ROI position): The purpose of setting the ROI position is to remove redundant data from the system and to increase the system computational speed. First, the initial detected line is set to the position where the vehicle is closer to the camera. Our experiment indicated that a substantial amount of unnecessary computation can be removed by setting the ROI position above the detected line (approximately at the bottom quarter of the



Fig. 3. (Color online) Lane line positions.

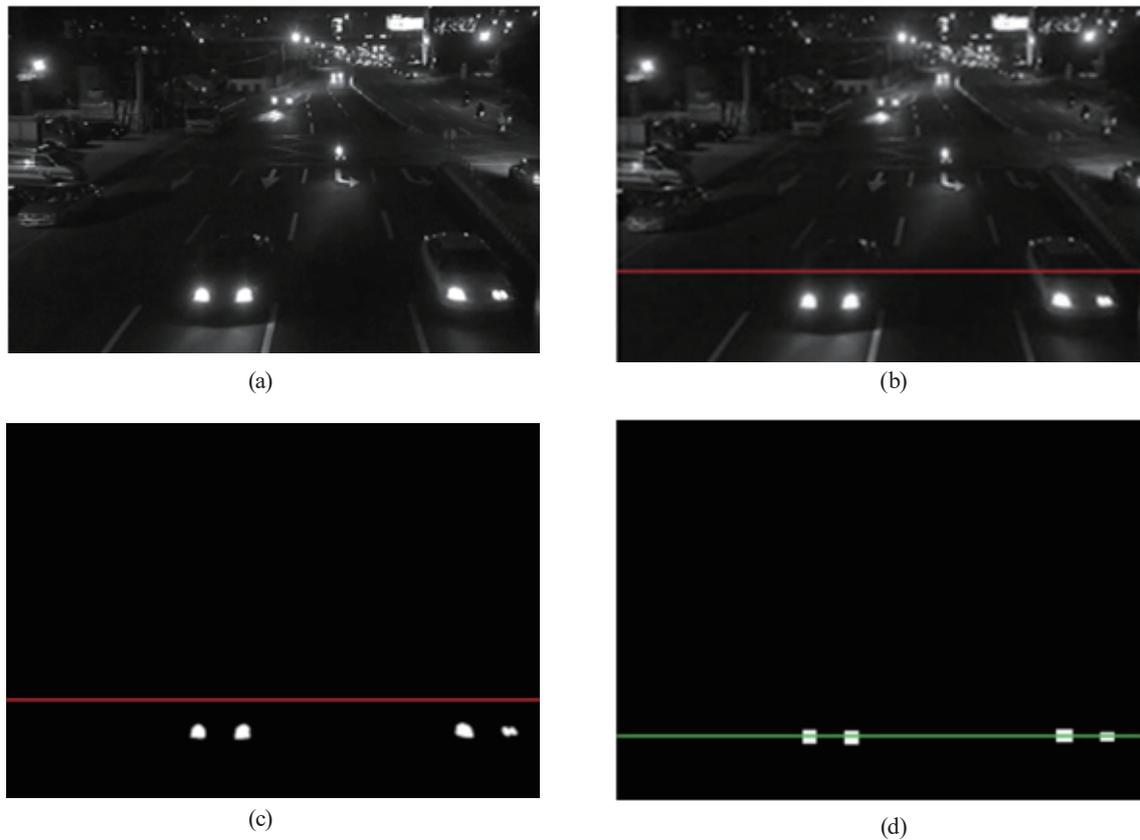


Fig. 4. (Color online) Preprocessing: (a) Grayscale image, (b) ROI, (c) binarization, and (d) vehicle light rectangularization (and line detection).

image), as shown in Fig. 4(b). While obtaining the ROI, an appropriate threshold is used to filter out low-energy light sources, so as to highlight the light sources of vehicle headlights.

Step 3 (Binarization of ROI): Otsu's dynamic threshold method is used to identify the binarization threshold value; this is carried out by performing repeated tests on public images and identifying an appropriate threshold that fulfills our requirements. The application of Eq. (1) in our experiments indicated that the use of the appropriate threshold ($T = 220$) in the binarization process allows for low-energy light sources to be filtered out, and that the conversion of grayscale images into binarized images helps highlight the light sources that originate from vehicle headlights, as shown in Fig. 4(c).

Step 4 (Rectangularization of vehicle lights and setting of detected line): The binarized but irregularly shaped white blocks are rectangularized by identifying their topmost, bottommost, leftmost, and rightmost points, and converting all values within this range to 1. Rectangularization will make it easier to accurately compute the proportions and spacing of vehicle headlights in the future. The position of the detected line is then set at the upper edge of the bottom 1/8 section, and on the basis of the lane line positions determined using a subroutine, the detected line is divided into four segments (i.e., Lanes 1, 2, 3, and 4), as shown in Fig. 4(d).

3.3 Detected line scanning and vehicle headlight recognition for lane-straddling vehicles

Segments 1 to 4 of the detected line, which correspond to lanes 1 to 4, are scanned from left to right. When a light source is detected, the said light source is recognized and its aspect ratio is obtained. If the aspect ratio of the light source meets the preset value range, it is recorded as a vehicle light and the scanning process will continue rightward. When another light source is detected and its aspect ratio also meets the preset value range, the spacing between the two vehicle lights is then computed. Should this spacing meet the preset value range, the presence of a vehicle is confirmed and the count increases by 1, as shown in Fig. 5.

Figure 6 shows a separate scan of an entire detected line. The points that overlap the lane line and detected line positions (as identified through a subroutine) are marked, and the detected line in question is then scanned and analyzed using the aforementioned method. When a vehicle is identified, it will be scanned from left (left vehicle light) to right (right vehicle light), and the count will increase by 1 if a marking point is detected. To prevent duplicate counts, the recorded data for each group are compared with those from the preceding group. The two groups of data will be considered to be from the same vehicle if they meet one of the following conditions: (1) the ratios of the left vehicle lights differ by a value of less than 0.5; (2) the ratios of the right vehicle lights differ by a value of less than 0.5; and (3) the distances between the two vehicle lights differ by less than 10. The final total number of vehicles is then sent to the Arduino platform to facilitate the control of the LED lights.

4. Experimental Results and Comparisons

4.1 Experimental data for vehicle detection

All gray level frames are used by transferring the RGB image to the YCbCr system. To verify whether the proposed algorithm is effective when applied to actual streets, in this work, we utilized nighttime traffic flow data that were captured from actual streets. The experimental



Fig. 5. (Color online) Four lanes.



Fig. 6. (Color online) Lane-straddling vehicle.

environment is established using an Intel Core i5 CPU@3.50 GHz, 16 GB RAM, and Microsoft Windows 10, and Matlab 2016a is chosen as the software development platform.

Figure 7 shows images of actual streets that were captured and measured in this work. Tables 1 to 5 contain objective data pertaining to the images, including image length, correction accuracy, and the computational time that the work achieved. These experimental data indicate that the use of two detected lines in a previous study to perform vehicle counting and avoid duplicate counts reduced the computational speed for that study.⁽⁷⁾ However, owing to its

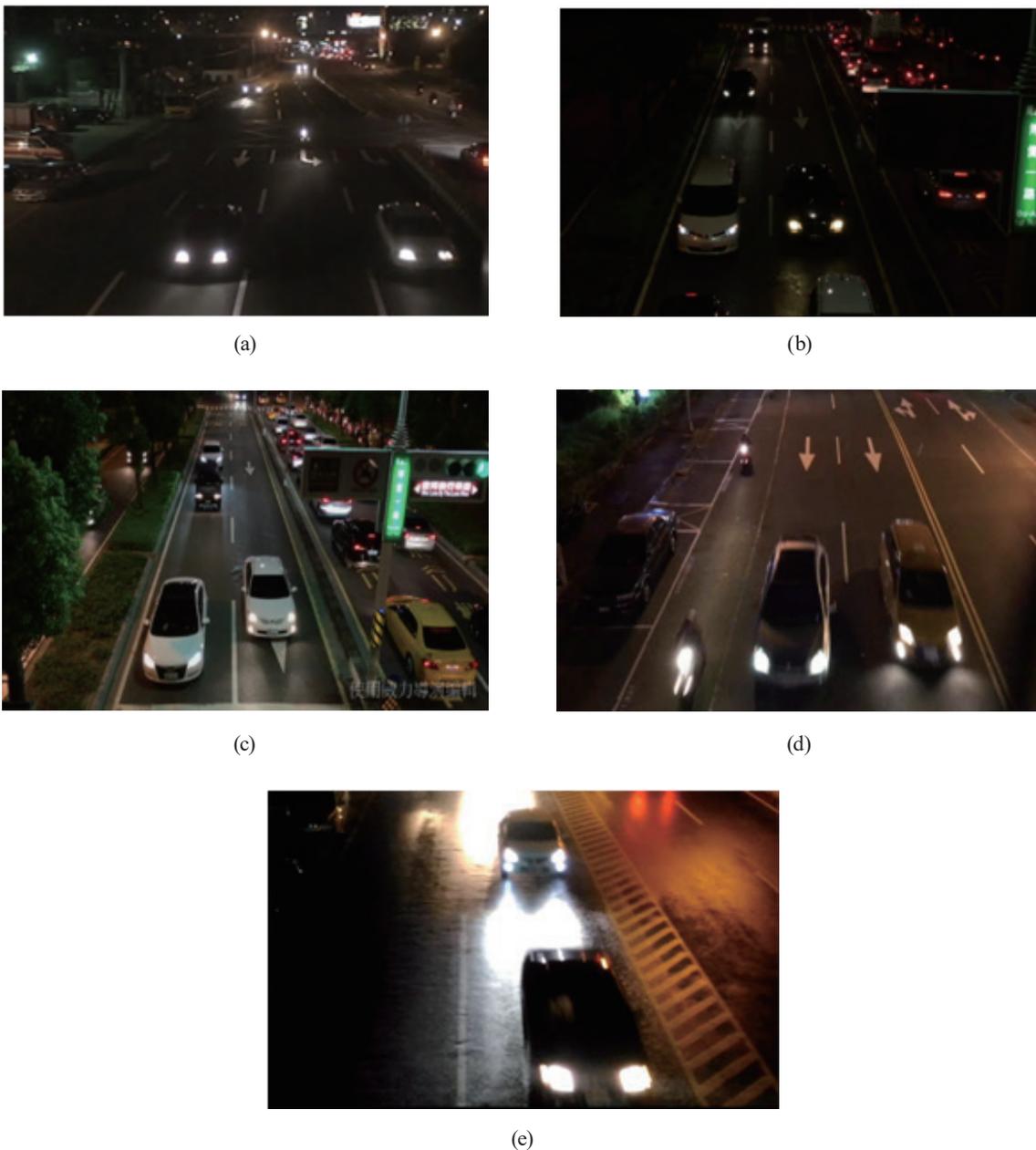


Fig. 7. (Color online) Test images captured from actual streets: (a) TestVideo01, (b) TestVideo02, (c) TestVideo03, (d) TestVideo04, and (e) TestVideo05.

Table 1
Experimental data for TestVideo01.

Resolution and time	704 × 528 (5''00')		
Methods	This work	Ref. 2	Ref. 7
Number of actual vehicles		43	
Number of vehicles detected	42	56	44
Correction rate	97.67%	67.44%	97.67%
Computing time	41.63 fps	23.98 fps	13.59 fps

Table 2
Experimental data for TestVideo02.

Resolution and time	720 × 480 (3''00')		
Methods	This work	Ref. 2	Ref. 7
Number of actual vehicles		58	
Number of vehicles detected	53	65	52
Correction rate	91.38%	89.23%	89.67%
Computing time	48.41 fps	20.94 fps	8.03 fps

Table 3
Experimental data for TestVideo03.

Resolution and time	720 × 480 (3''05')		
Methods	This work	Ref. 2	Ref. 7
Number of actual vehicles		79	
Number of vehicles detected	78	7	77
Correction rate	98.73%	8.86%	97.47%
Computing time	48.05 fps	17.20 fps	6.41 fps

Table 4
Experimental data for TestVideo04.

Resolution and time	720 × 480 (3''05')		
Methods	This work	Ref. 2	Ref. 7
Number of actual vehicles		44	
Number of vehicles detected	42	123	44
Correction rate	95.46%	N/A	100%
Computing time	39.23 fps	17.32 fps	7.92 fps

Table 5
Experimental data for TestVideo05.

Resolution and time	720 × 480 (5''00')		
Methods	This work	Ref. 2	Ref. 7
Number of actual vehicles		35	
Number of vehicles detected	41	423	20
Correction rate	85.36%	N/A	57.14%
Computing time	42.85 fps	18.07 fps	6.81 fps

application of Otsu's method for binarization, another study encountered issues related to the misjudgment of a large volume of noise as vehicle lights, which reduced the accuracy rates achieved by that study. This issue is particularly pronounced in the results in Tables 3 to 5, which could be attributed to the fact that the corresponding environments suffered from a greater degree of light source interference that substantially reduced the accuracy rate.⁽²⁾ In Table 2, the drop in detection rate was a result of overly dim vehicle lights.

In Table 5, the lower accuracy rate was caused by the vehicle light reflections that occurred under wet road conditions. However, the method proposed in this work achieved a higher average nighttime correction rate of 93.63% and higher, as well as an average computing time of 44.03 frame per second (fps). However, note that a drop in correction accuracy (i.e., misjudgments) also occurred when vehicle features became less clear owing to various circumstances, including two scooters traveling parallel, overly dim vehicle lights, vehicle lights that were switched off, and various weather conditions.

4.2 Results related to control of simulated LED street lights

In this work, we proposed the monitoring of nighttime traffic volume as a means of implementing the automated control of simulated LED street lights, as shown in Fig. 8. It was

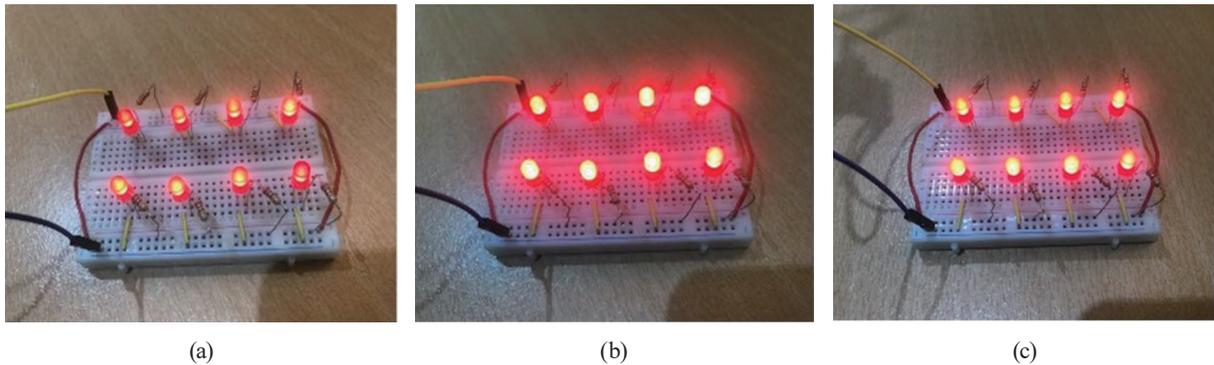


Fig. 8. (Color online) Simulated LED street lights: (a) initial state, (b) high LED power (brighter) when the traffic volume is high, and (c) low LED power (dimmer) when the traffic volume is low.

assumed that there were no vehicles passing through at the start of the experiment, and that the LED lights were adjusted to a preset state. After some time had passed, the methods proposed in this study were used to compute the traffic volume and automatically adjust the LED light brightness. The adjustment is performed as follows: (1) the LED lights are automatically dimmed when a low traffic volume is detected; (2) the LED lights are automatically brightened when a high traffic volume is detected; and (3) the LED lights are maintained in their preset state when no passing vehicles are detected.

5. Conclusions

In this work, we proposed a multifeatured technique that could be used to perform real-time vehicle detection at nighttime. The application of the Hough line transform to detect lane lines addressed the problem of two vehicles being misjudged as one when they move in parallel and exhibit similar characteristics, and improved the system vehicle detection rates and real time processing capabilities. The experimental results showed that the nighttime correction rate and computing time of the technique reached 93.63% and 44.03 fps, respectively. However, the issues that future studies could address include the following: handling of a small number of vehicles that do not switch on their headlights at night and vehicle lights with unusual designs. These issues might lead to misjudgments that will reduce the system accuracy. If the method proposed in this work is eventually implemented in actual highway environments and integrated into the future network of LED street lights, it would enable the control of street lights and could potentially reduce electricity consumption by roughly 30% on a daily basis.

References

- 1 K.-F. Yeh: Master Thesis, National Sun Yat-Sen University (2012).
- 2 K.-W. Chao: Master Thesis, National Changhua University of Education (2013).
- 3 H.-L. Wu: Master Thesis, National Cheng Kung University (2011).
- 4 Y. Yuan, Y. Zhao, and X. Wang: *Int. Conf. Image and Vision Computing* (2013) 453–458.
- 5 W. Zhang, Q. M.-J. Wu, G. Wang, and X. You: *IEEE Trans. Intell. Transp. Syst.* **13** (2012) 140.

- 6 J.-M. Guo, C.-H. Hsia, K. S. Wong, J.-Y. Wu, Y.-T. Wu, and N.-J. Wang: *IEEE Trans. Veh. Technol.* **65** (2016) 4023.
- 7 C.-H. Hsia and Y.-K. Lin: *J. Chung Cheng Inst. Technol.* **46** (2017) 1.
- 8 A. Borkar, M. Hayes, M. T. Smith, and S. Pankanti: *IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems* (2009).
- 9 H.-Wang and Q. Chen: *IEEE Intelligent Transportation Systems Conf.* (2006).
- 10 K. Robert: *IEEE Conf. Intelligent Transportation Systems* (2009) 285–290.
- 11 D. Jurić and S. Lončarić: *Int. Conf. Connected Vehicles and Expo* (2014) 655–660.
- 12 N. Markuš, M. Frljak, I. S. Pandžić, J. Ahlberg, and R. Forchheimer: *arXiv preprint arXiv:1305.4537* (2014).
- 13 U. Kumar: *Int. Conf. Image and Vision Computing* (2013) 214–219.
- 14 J. Chen, J. Chen and F. Gu: *Int. Conf. Intelligent Human-Machine Systems and Cybernetics* (2015) 480–483.
- 15 J.-W. Park and B. C. Song: *Int. Conf. Electronics, Information, and Communications* (2016).
- 16 H.-T. Chen, Y.-C. Wu, and C.-C. Hsu: *IEEE Sens. J.* **16** (2016) 120.
- 17 J. Zhu, H. Zeng, Y. Du, Z. Lei, L. Zheng, and C. Cai: *IEEE Access* **6** (2018) 43724.
- 18 V. Priyashman and W. Ismail: *IEEE Sens. J.* **19** (2019) 1400.
- 19 O. Khin, M. Phothisonothai, and S. Choomchuay: *Sens. Mater.* **30** (2018) 2247.
- 20 Y. Kong and S. Pan: *Sens. Mater.* **31** (2019) 1917.