Vehicle Detection Algorithm for Applications Pertaining to License Plate Recognition

Bing-Fei Wu and Meng-Liang Chung*

Institute of Electrical Control Engineering, National Chiao Tung University

ABSTRACT

The technology of license plate recognition (LPR) is widely applied to parking lot management systems, intelligent transportation systems, and electronic toll collections. The goal of LPR is to identify license plates quickly and accurately. Therefore, many researchers have proposed methods for achieving this goal. This paper proposes an algorithm that uses information regarding the location of the vehicle to trigger the camera to capture an image of the vehicle. This not only achieves the goal of detecting an image in real time, but also obtains the best image with the license plate as a triggering image. The proposed algorithm does not require additional hardware, and facilitates the precise retrieval of an image that both contains a vehicle and represents the best image from a series of images for recognizing the characters on license plates. This significantly reduces the cost of hardware and is much easier and cheaper to maintain than the traditional detection methods. Experimental results show that the detection success rate of the proposed algorithm reached 92% during the daytime and nighttime. Despite diverse weather conditions, moreover, the detection success rate of the proposed algorithm reached 91%. An extensive vehicle-detection test demonstrates that the proposed algorithm is reliable and accurate.

Keywords: license plate recognition, image trigger, moving vehicle.

應用於智慧型車輛的視覺感測系統

吳炳飛 鐘孟良*

國立交通大學電控工程研究所

摘 要

車牌辨識系統廣泛應用於停車場管理系統、智能運輸系統及電子收費系統。因此，許多學者提出不同的方法想要達到此目的。一般而言，目前並沒有針對擷取車輛車牌的的最佳位置影像，提供給車牌辨識系統做進一步辨識的研究，本文即針對此問題提出一個利用攝影機擷取車輛車牌的最佳位置的演算法。本系統不僅滿足即時偵測的目標，且可在連續影像中獲得一張具有最佳車輛車牌的觸發影像，並且無需額外架設特別的硬體裝置，本演算法提出精確擷取具有最佳車輛車牌的觸發影像，並提供給車牌辨識系統進行車牌辨識。與傳統的偵測方法相比，可大量省下硬體的成本及維護費用。實驗結果顯示，本演算法在白天及夜間的觸發成功率可達 92%以上；在不同天候下的觸發成功率可達 91%以上。藉由一週的長時間測試，實驗結果顯示本演算法是穩定且傑出的演算法。

關鍵詞：車牌辨識，觸發影像，移動車輛

文稿收件日期 105.4.19;文稿修正後接受日期 105.6.3; *通訊作者
Manuscript received April 19, 2016; revised June 3, 2016; * Corresponding author

I. INTRODUCTION

License plate recognition (LPR) refers to image processing technology used to identify vehicles by their license plates. This technology is widely applied to various security and traffic applications, such as parking lot management systems, intelligent transportation systems, and electronic toll collections. To identify license plates quickly and accurately, scholars have proposed several methods to reduce the time required to identify license plates correctly. For example, Gao et al. [1] proposed a novel framework for vehicle make and model recognition (MMR), using local tiled deep networks. The frontal views of vehicle images are first extracted and fed into the local tiled deep network for training and testing. The accuracy of vehicle MMR was approximately 98%. Among the methods used for comparison—including LBP, LGBP, SIFT, Linear SVM, RBM, CNN, TCNN, LTCNN, and LTCNN with HOG—the computational times ranged between 539 ms and 3842 ms. The MMR proposed by Gao et al. (viz., LTCNN with GOG) obtained the highest accuracy. Zheng et al. [2] proposed a method for detecting characters from an input image. Their method involved scanning the image with a movable window to detect characters. The detection speed was around two seconds. Li [3] proposed a geometric framework for rectangular shape detection. This method detects the frames of license plates, with a processing time of approximately 1.1 seconds per image. Wijjnhoven et al. [4] detected the frames of license plates with sliding analysis windows. With their method, the time required to detect the license frame was around 0.3 seconds. Ashtari et al. [5] proposed a license plate recognition system that identifies the characters of license plates by using color features. The recognition system required around 1.59 seconds per image. Thooyamani et al. [6] proposed an algorithm using a neural network module to identify the characters of license plates. Their algorithm required approximately two seconds per image. Chang et al. [7] proposed a method that uses license plate location and identification models to detect characters on license plates. Their method required approximately 2.4 seconds per image. Hsu et al. [8] proposed solution consists of three modules for plate detection, character segmentation, and recognition. The recognition system longest processing time, namely 0.21, 0.26, and 0.32 seconds per image on the access control, law enforcement, and road patrol subsets. Although all of the above methods attempted to improve the time required to identify the characters of license plates, none of these methods achieved the standard set by the National Television System Commission (NTSC) of 30 frames transmitted each second. Even in different platforms, it is difficult to achieve this standard.

Some approaches combine segmentation and recognition methods for detecting the characters on license plates. Li [9] proposed an algorithm to segment the characters of license plates using the local gradient quantity, showing a detection time of 0.573 seconds. Deb et al. [10] utilized the hue saturation intensity (HSI) color model and an adaptive threshold for the candidate areas of license plates. The detection speed of characters with their method was around one second. Comell et al. [11] used optical character recognition (OCR) to recognize the characters of license plates. Their method's execution speed was 1.1 seconds.

Some studies have adopted hardware devices to trigger a camera to detect a moving vehicle. Wu et al. [12] utilized an infrared sensing device to detect whether the vehicle is in the detection area. When a vehicle moves into the detection area, the infrared sensing device immediately triggers the camera to capture the images of vehicles and identify the license plates. However, the costs of hardware equipment and maintenance are typically high with such methods, making them practically unfeasible.

The use of images for LPR is a viable approach. To our knowledge, however, there is no related application regarding the use of images to detect the location of a vehicle. To detect vehicles, the most common methods are temporal difference (TD) and background subtraction (BS). Regarding TD, Nakanishi [13] compared two images to detect the moving object when the difference was greater than a threshold. This method quickly detects moving objects, but detection errors are common whenever the object is static or moving slowly. Li [14] utilized the difference between adjacent images to converge the background. However, this method risks regarding slow-moving objects as the background, resulting in detection errors. Moreover, errors were common when detecting subsequent moving objects. From the above descriptions, we can conclude that using TD for LPR can lead to errors. In addition, the sensitivity of the LPR camera is high, such that the brightness of the image changes with the luminance of the objects. Thus, when using TD to obtain a background image, fragmented images can result.

On the other hand, BS uses the difference between the background image and the input image in order to segment objects through the background image captured by cameras. Because BS includes a background image for comparison, it can segment objects even when an intrusive object is static. However, when a BS system operates over a protracted period of time, the background image must be updated simultaneously in order to segment the objects correctly. In addition, BS encounters problems with the initial background convergence. There are two ways to retrieve the initial background: the first involves using the average, and the second involves using a probability model. Lipton et al. [15] retrieved images during a period of time and calculated the average pixels as the background. However, errors occur when a moving object is captured during the retrieval process. Chiu et al. [16] processed background detection using a probability model. However, their method risks detecting the wrong background when large vehicles appear in the detection screen. Wu et al. [17] proposed a BS method for detecting moving vehicles. Because the stability of the background must be maintained, a specific camera with less gain is needed in order to acquire a stable background. Methods related to BS are thus unsuitable for LPR. The main reason for this is that, although TD can detect the entire vehicle, the gain from the camera causes the brightness of the images to change considerably, such that there is no way for TD to obtain a stable and effective static background image. In addition, although methods that use the difference in adjacent images to converge the background are effective, they are disadvantageous insofar as the probability of erroneous judgments is high. When a moving vehicle appears in the detection screen, too many images can be regarded as trigger images. This impedes license plate recognition systems from real-time detection.

To solve the above two problems, this paper proposes an algorithm for detecting moving vehicles using the concepts of both BS and TD. The proposed algorithm first detects a static background image using TD after initiating the system. Then, it detects the locations of the front and rear of the vehicle. When no vehicle is detected, the algorithm retrieves one image as the background image and utilizes this background image to conduct BS. This differs from other methods, which perform convergence in order to obtain the background image first. During the BS process, the algorithm uses a threshold to compensate for changes in brightness. Then, it utilizes the information regarding the location of the vehicle to trigger the camera and capture an image of the vehicle. The algorithm can meet the goal of detecting images in real time. In addition, it detects images with the best identification area to act as a trigger from continuous images. Without any requirements for additional hardware, the proposed algorithm triggers the camera to capture an image.

In traditional LPR system, the cost of hardware is high. This manuscript proposed the use of images for detecting the location of a vehicle and further obtaining the best image for the LPR system. This significantly reduces the cost of hardware and is much easier and cheaper to maintain than the traditional detection methods. This results in a significant reduction in the cost of hardware, and it is much easier and cheaper to maintain than traditional detection methods. Most importantly, the algorithm precisely retrieves the best image from a series of images that contain the vehicle. This selected image is used for further processing in order to recognize the characters of license plates. The accuracy of LPR is thus considerably improved.

This paper is organized as follows. Section 2 presents the proposed algorithm. Section 3 presents the experimental results and a discussion of these results. Finally, Section 4 concludes the paper.

Ⅱ. METHOD

Most studies propose methods regarding the correct recognition of the characters of license plates. However, to the best of our knowledge, there is no related research regarding the retrieval of the best image from a series of images of the vehicle, such that this best image can subsequently be used to better recognize the characters of license plates. In order to accelerate the recognition speed and improve the recognition rate of an LPR system, this paper proposes an algorithm to effectively acquire the best images of a vehicle with a license plate for the LPR system. The main processes of the proposed algorithm consist of dynamic vehicle detection, vehicle and background renewal, and vehicle location analysis. A flowchart for the proposed algorithm is shown in Figure 1.

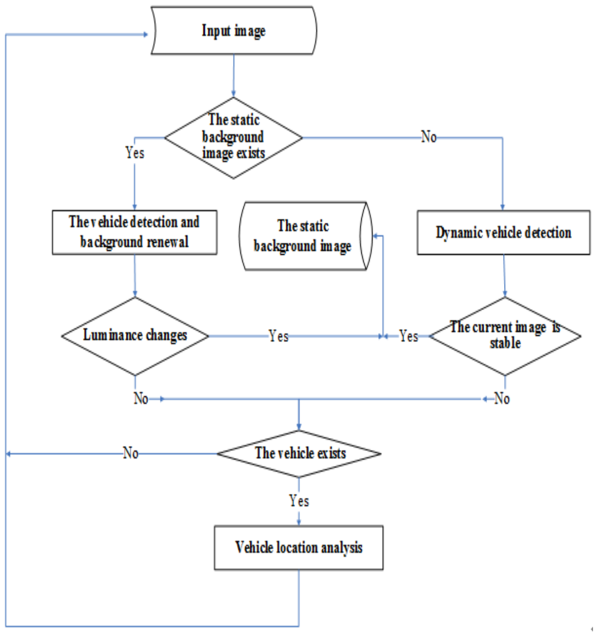


Fig.1. Flowchart of the proposed algorithm.

When the algorithm initiates, there is no static background image. Therefore, the algorithm utilizes dynamic vehicle detection to analyze the input image sequentially, and, further, to judge whether it is stable. When the current image is judged to be stable, it is stored as the static background image. If not, the algorithm further confirms whether a vehicle exits in the current image. When a vehicle is detected in the current image, the algorithm utilizes vehicle location analysis to obtain the best image of the vehicle's license plate. If a static background image is obtained during dynamic vehicle detection, the algorithm proceeds to the subsequent processes: viz., vehicle detection and background renewal. At this time, the process utilizes the static background image for comparison with the input image in order to check whether there is any obvious change in luminance. If the change in luminance exceeds a predefined threshold and there is no vehicle in the input image, the input image is stored as the static background image—replacing the original background image. If the change in luminance does not exceed the threshold, the algorithm next judges whether there is a vehicle in the image. When a vehicle is detected, then the algorithm employs vehicle location analysis in order to obtain the best image of the vehicle's license plate for the LPR system. When no vehicle is detected, the algorithm returns to the beginning of the flowchart and repeats the process. Details for these processes are described in the following subsections.

2.1 Dynamic vehicle detection

The first process of the algorithm is dynamic vehicle detection. The main purpose of this process is to obtain a static background image and to detect whether a dynamic vehicle exists. The steps for doing so are as follows.

Step 1: Subtract two continuous images.

This step is mainly used to retrieve two continuous images, as shown in Figure 2a and Figure 2b, and to obtain the results of the subtraction of these two images.

Step 2: Generate a histogram of grayscale difference between these two continuous images, and find the segmentation threshold.

After Step 1, a histogram of grayscale differences is generated for these two continuous images. Generally speaking, when there is no moving object in two continuous images, the grayscale differences are very small, with most located at 0, as shown in Figure 2c. In the histogram, Step 2 derives the minimum valley of the histogram, which is used as the automatic segmentation threshold, T_1.

Step 3: Obtain the foreground image using the threshold.

After deriving the threshold, Step 3

involves obtaining the foreground using this threshold. The foreground image is the image with pixels that are more than or equal to the threshold. All of these pixels are set to 255. Conversely, for images with pixels less than the threshold, these pixels are set to 0. Subsequently, this step records the threshold.

Step 4: Group the components.

Before grouping the components, dilation and erosion are applied to the foreground image in order to reduce noise and smooth the contour of the image. Then, connected-component labeling is used to group the pixels of the foreground image into components. Smaller components whose widths are less than a tenth of the width of the image are removed. The reason for removing such components is that it is impossible for these components to be part of the license plate. It is assumed that W_c represents the width of the connecting component, and that W_{min} is the width of the license plate. All of the connecting components shall satisfy the following condition:

$$W_c \geq \frac{1}{10} W_{min} \tag{1}$$

The locations of objects that are at least a tenth of the width of the input image are recorded in the input image.

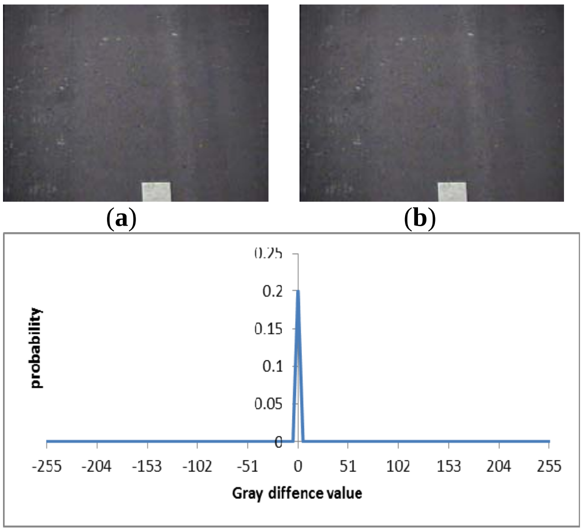
In the actual image, the image width of license plate is less than a tenth of the image width, the spacing between the characters is very close and difficult to identify.

Step 5: Judge whether the component exists.

After removing all of the small components, the input image is stored as the static background image when there are no components left. When large components remain, this suggests that the front of a vehicle might appear in the foreground image. In this case, the foreground image must be processed further.

Step 6: Find the front area of the vehicle.

In order to detect whether a large component is indeed the front of a vehicle in the foreground image, this step accumulates the sum of the horizontal pixels in the foreground image and attempts to find the location of the maximum sum. The location of the maximum sum represents the suspected lower edge of the vehicle. Upon obtaining the location information, it is transmitted to vehicle location analysis for further processing.



(a) (b)

(c)

Fig.2. Result from finding the automatic segmentation threshold, (a) input image 1, (b) input image 2, (c) histogram of grayscale differences.

2.2 Vehicle detection and background renewal

After obtaining the static background image, the algorithm proceeds to the vehicle detection and background renewal process. The purpose of this process is to detect images with vehicles and find the location of the lower edge of those vehicles. After obtaining location information, the process transmits this information to vehicle location analysis for further processing. The detailed steps of the process are as follows.

Step 1: Subtracting the static background image from the input image.

This step obtains the results from subtracting the static background image from the input image.

Step 2: Generate a histogram of grayscale difference between the static background image and the input image, and find the segmentation threshold.

After Step 1, a histogram of grayscale differences is generated between the static background image (Figure 3a) and the input image (Figure 3b). Using the histogram (Figure

3c), this step next determines the minimum valley, which is used as the automatic segmentation threshold, T_2 .

Step 3: Obtain the foreground image using the automatic threshold.

After deriving the automatic threshold, Step 3 involves obtaining the foreground using the automatic threshold. The foreground image is the image with pixels that are more than and equal to the automatic threshold. All of these pixels are set to 255. Pixels less than the threshold are set to 0. Subsequently, the foreground image is obtained, as shown in Figure 3d. Then, the step records the automatic threshold.

Step 4: Group the components.

Before grouping the components, dilation and erosion are applied to the foreground image in order to reduce noise and smooth the contour of the image. Then, connected component labeling is used to group the pixels of the foreground image into components. Components whose widths are less than a tenth of the width of the image are removed. The reason for removing these components is that it is impossible them to be part of the license plate. Moreover, all of the connecting components must satisfy Eq. (1). The locations of objects at least a tenth of the width of the input image are recorded in the input image.

Step 5: Judge whether the static background image must be renewed.

This step involves judging whether the static background image must be renewed. The criteria for this are, first, that the amount of threshold change exceeds 30, and, second, that the images are continuous (up to 150 images). When these criteria are satisfied, the static background image needs to be renewed. The current input image is then stored as the static background image. When these criteria are not satisfied, on the other hand, the algorithm proceeds to Step 6.

Step 6: Perform Sobel vertical edge detection

This step involves performing Sobel vertical edge detection, using Eq. (2) to find the vertical edges of the foreground image. Here, δ denotes the down-sampling factor:

$$DSE_H(x,y) = |DS(x,y-\delta)| - |DS(x,y+\delta)| \quad where\ x = 1,2,.... \ and\ y = 1,2,.... \qquad (2)$$

The result is shown in Figure 3e.

Step 7: Perform the vertical and horizontal projections of the object.

After obtaining the vertical edges, the vertical projection is applied to determine the number of pixels in each column of the foreground image. A maximum of two numbers is possible to the left and right side of the object. On the same principle, the horizontal projection is applied to determine the number of pixels in each row of the foreground image. The location of the maximum number, which represents the suspected lower edge of the front of the vehicle, is recorded and transmitted for vehicle location analysis.

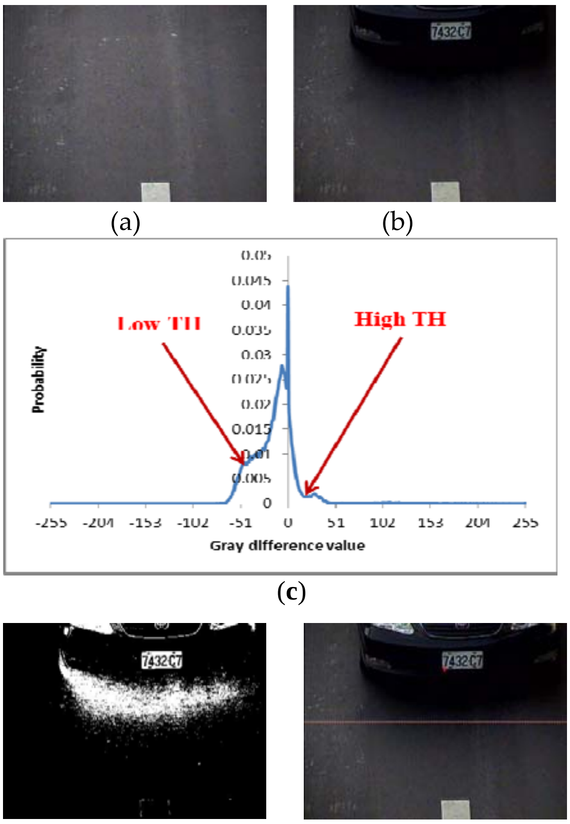


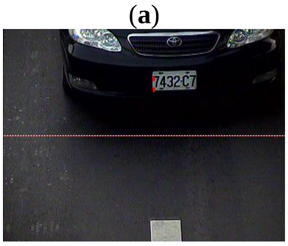
Fig.3. Results from moving vehicle detection, (**a**) static background image, (**b**) input image, (**c**) histogram of grayscale differences, (**d**) foreground image, (**e**) potential area of the license plate.

2.3 Vehicle location analysis

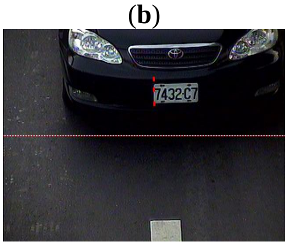
Vehicle location analysis involves using the vehicle location in each image to find the best image of license plate for the LPR system. This process consists of vehicle tracking and the triggering process.

First, information is acquired regarding the vehicle locations and the lower edges of the vehicles—from the dynamic vehicle detection process, and the vehicle detection and background renewal process, respectively. When the lower edges of the vehicles appear in the region of interest (ROI), the vehicle tracking process begins tracking the vehicle and recording its location. In this paper, the size of ROI is defined as the upper half of the image, as seen in Fig. 4. The triggering line is the red dotted line located on the axis of Y/2. The algorithm continues to track and record the location of the lower edge of the vehicle, as shown as Figures 4a—4f. After the lower edge of the vehicle passes through the triggering line, the triggering process analyzes all of the recorded locations of the lower edge of the vehicles and obtains the image that is closest to the triggering line—the axis of Y/2. This image represents the best image for the LPR system, as shown Figure 4f.

(**a**)

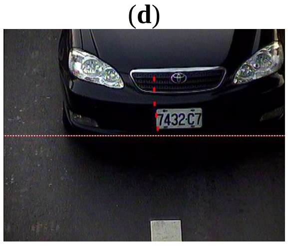


(**b**)

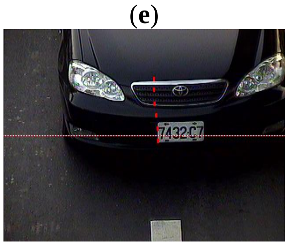


(**c**)

(**d**)



(**e**)



(**f**)

Fig.4. Results from detecting the best image of the license plate, (**a**) input image frame n, (**b**) input image frame n+1, (**c**) input image frame n+2, (**d**) input image frame n+3, (**e**) input image frame n+4, (f) input image frame n+5.

Ⅲ. RESULTS and DISCUSSION

The proposed algorithm must be work together with other hardware, including an image processing unit and a digitalization unit. To evaluate the proposed algorithm, we used an image processing unit consisting of a 1/3 inch Charge Coupled Device (CCD) license plate capture camera with a 5-50 mm lens. After capturing images with this camera, all analog signals were sent to a digitalization unit with the image capture card. In addition, the image capture card can process 30 images per second in real time. All captured images were then sent to the algorithm unit for further processing in order to obtain the best image, which was used as the triggering image. Images showing the installation of the system are shown in Figure 5. The red square frame in Figure 5b shows that the height of the system is 4.6 m. The green square frame in Figure 5b shows the system which has been set up.

(a) (b)

Fig.5. System installation, (**a**) system under construction, (**b**) system after construction.

The hardware platform comprised an Intel Core i7-7490 processor with 4 GB of RAM and a PCIe-RTV24 frame grabber produced by ADLINK. The PCIe-RTV24 provides a four-channel digitizer vision stream for real-time image acquisition. For each channel, the acquisition speed reached up to 30 images per second. The program development environment was Visual Studio.Net 2013. When a static background image was derived, the average time to obtain the best triggering image was approximately 2 ms. When there was no static background image, the time required to calculate the gray value differences using two continuous frames and the dynamic segmentation threshold was approximately 1 ms. With a static background, the time for detecting moving objects was approximately 3 ms. Subsequently, the time required to detect moving vehicles was around 1 ms. Therefore, the average time for processing each frame was less than 6 ms.

In order to obtain more data to demonstrate the superiority of the proposed algorithm, a camera was set up along a footbridge 4.6 m above the road. The depression angle of the camera was set to approximately 15 degrees, and the lens of the camera was aimed toward the front of the vehicles for further detection. The proposed algorithm was optimized such that it could execute uninterrupted 24-hour vehicle detection in real time. In addition, we tested the algorithm's accuracy under different situations. Table 1 shows the detection success rate of vehicles during daytime and nighttime. Daytime was set between 6 a.m. and 6 p.m. (i.e., 12 hours) and nighttime was set between 6 p.m. and 6 a.m. (i.e., 12 hours). Although the luminance of the nighttime image was low, the detection success rate for vehicles during daytime and nighttime reached 92%. Figure 6 shows the detection results during daytime and nighttime.

Table 1. Detection results during daytime and nighttime.

Time	Total vehicle	Triggering vehicle	Trigering rate
Daytime	1353	1271	94%
Nighttime	850	782	92%

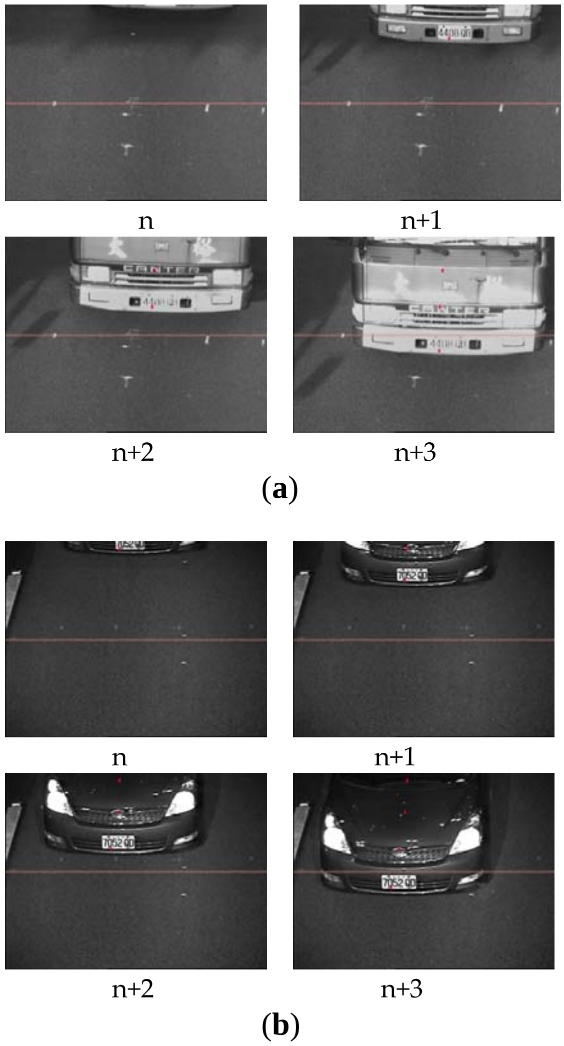
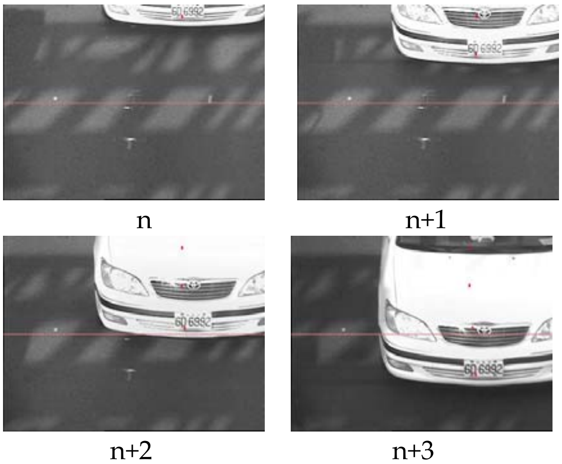


Fig.6. Detection results, (a) detection results during daytime, (b) detection results during nighttime.

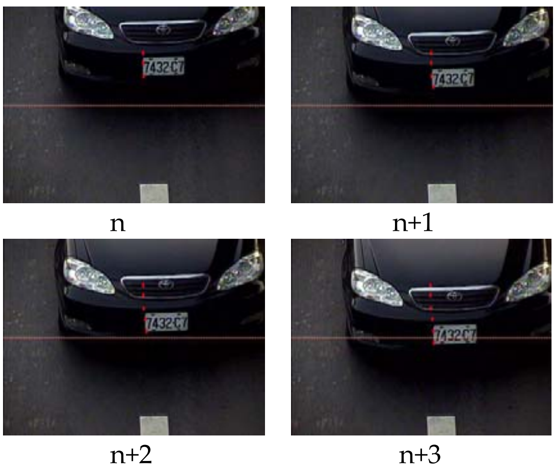
The quality of the images captured by the camera was poor on rainy days. However, the proposed algorithm effectively overcame such weather factors. Under different weather conditions, the detection success rates of vehicles reached 91%. The detection results are shown in Table 2. Figure 7 shows the detection results on sunny day, cloudy day, and rainy day.

Table 2. Detection results on sunny days, cloudy days, and rainy days.

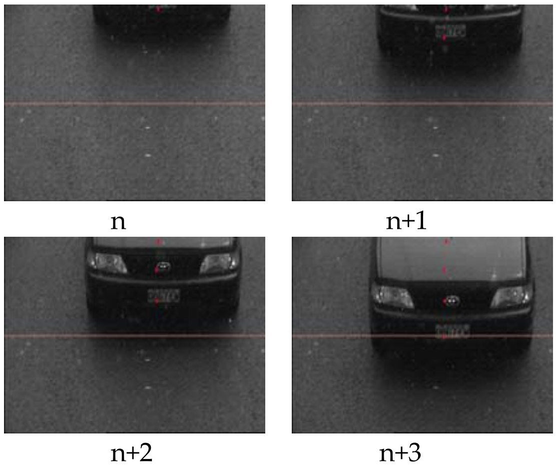
Time	Total vehicle	Triggering vehicle	Trigering rate
Sunny day	635	610	96%
Cloudy day	420	392	93%
Rainy day	535	486	91%



(a)



(b)



(c)

Fig.7. Detection results, (**a**) detection results on sunny day, (**b**) detection results on cloudy day, (c) detection results on rainy day.

Moreover, in order to measure the performance of the proposed algorithm, a one-week vehicle detection test was conducted. The detection results are shown in Table 3. The detection success rate for vehicles reached 94% during this experiment. In addition to obtaining the triggering images for the LPR system, the number of vehicles each day could be estimated with the algorithm. This information can be provided to authorities to further improve traffic flow.

Table 3. Detection results on weekdays and weekends.

Time	Total vehicles	Triggering vehicles	Detection success rate
Monday	2132	2012	94%
Tuesday	2068	1955	95%
Wednesday	1887	1801	95%
Thursday	2006	1908	95%
Friday	2388	2238	94%
Saturday	2260	2125	95%
Sunday	2632	2501	95%

Ⅳ. CONCLUSIONS

In this paper, a vehicle detection algorithm was proposed. The proposed algorithm precisely retrieves the best image of a vehicle’s license plate from a series of images in real time. The obtained triggering images are then further processed by an LPR system to recognize the characters on license plates. Because these triggering images represent the best images of the license plates, the proposed algorithm considerably improves the accuracy and speed of an LPR system. Without any special requirements for additional hardware, the algorithm significantly reduces the costs of hardware, and it is much easier and cheaper to maintain than traditional detection methods. Experimental results show that the detection success rate of the proposed algorithm during daytime and nighttime reached 92%. Under adverse weather conditions, the detection success rate of the proposed algorithm reached 91%. The results from a week-long vehicle detection test demonstrated the reliability and performance of the proposed algorithm.

REFERENCES

[1] Gao, Y., Lee, H.J. Local tiled deep networks for recognition of vehicle make and model. Sensors, 2016, 16, 226.
[2] Zheng, D., Zhao, Y., Wang, J. An efficient method of license plate location. Pattern Recognition letters, 2005, 26, 2431–2438.
[3] Li, Q. A geometric framework for rectangular shape detection. IEEE Trans. Image Process. 2014, 23, 4139–4149.
[4] Wijnhoven, R.G.J., With, P.H.N. Identity verification using computer vision for automatic garage door opening, IEEE trans. Consum. Electron., 2011, 57, 906–914.
[5] Ashtari, A.H., Nordin, M.J., Fathy, M. An Iranian license plate recognition system based on color features. IEEE Trans. Intell. Transp. Syst., 2014, 15, 1690–1704.
[6] Thooyamani, K.P., Khanaa, V., Udayakumar, R., Middle-East Journal of Science Research. 2013, 18, 1768-1774.
[7] Chang, S.-L., Chen, L.-S., Chung, Y.-C., Chen, S.-W. Automatic license plate recognition. IEEE Trans. Intell. Transp. Syst., 2004, 15, 42–53.
[8] Hsu, G.-S., Chen, J.-C., Chung Y.Z. Application-Oriented License Plate Recognition, IEEE Transactions on Vehicular Technology, 2013, 62(2), 552-561.
[9] Li, B., Tian, B., Li, Y., Wen, D. Component-based license plate detection using conditional random field model. IEEE Trans. Intell. Transp. Syst., 2013, 14, 1690–1699.
[10] Deb, K., Jo, K. Segmenting the license plate region using a color model. Pattern recognition, Yin, P.; Publisher: InTech, Rijeka, Croatia, 2009; pp. 401–418.
[11] Comelli, P., Ferragina, P., Granieri, M. N., Stabile, F. Optical recognition of motor vehicle license plates. IEEE Trans. Veh. Technol., 1995, 44, 790–799.
[12] Wu, P.-F., Lin, S.-P., Chiu, C.-C. Extracting characters from real vehicle license plates out-of-doors. IET Comput. Vis., 2007, 1, 2–10.
[13] Nakanishi, T.; Ishii, K. Automatic Vehicle Image Extraction Based on Spatio-Temporal Image Analysis. *IAPR* International Conference on Computer Vision and Applications, 1992, 1, 500–504.
[14] Li, C.; Ikeuchi, K.; Sakauchi, M. Acquisition of Traffic Information Using A Video Camera with 2D Spatio-Temporal Image Transformation Technique. IEEE Int. Conf. on Intell. Transp. Syst., 1995, 634–638.
[15] Lipton, A. J.; Fujiyoshi, H.; Patil, R. S. Moving Target Classification and Tracking from Real-Time Video. IEEE Workshop on Appl. of Comput. Vis., 1998, 8–14.
[16] Chiu, C. C.; Ku, M. Y.; Liang, L. W. A Robust Object Segmentation System Using a Probability-Based Background Extraction Algorithm. IEEE Trans. Circuits Syst. Video Technol., 2010, 20, 518–528.
[17] Wu, B. F.; Juang, J. H. Adptive Vehicle Detector Approach for Complex Environments. IEEE Int. Conf. on Intell. Transp. Syst., 2012, 817–827.