International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 11 (2015) pp. 28195-28210

http://www.ripublication.com

Real-Time Vehicle License Plate Detection and Recognition System

Kumar Rajan[1] V. Laxmi[2] D. K. Mohanta[3] S. Das[4]
[1]Research Scholar, Dept. of EEE, BIT, Mesra, Ranchi, India
[2]Associate Professor, Dept. of EEE, BIT, Mesra, Ranchi, India
[3]Professot, Dept. of EEE, BIT, Mesra, Ranchi, India
[4]Assistant Professor, Dept. of EEE, VVIT, Purnea, India

Abstract

In the past few years, numbers of vehicle have increased worldwide, particularly in India. Therefore, managing traffic is a very big challenge in India. The purpose of this work is to develop a laboratory model of real-time application for detecting and recognizing the vehicle license plate. The system can be installed at a toll gate or restricted entry area or at a border crossing. The system comprises of a computer with a video camera, which captures video frames including visible vehicle license plate and processes them. Once the vehicle license plate is recognized through the system, the characters are recognized and displayed in the graphical user interface (GUI) of LabVIEW and checked using the database. The focus is on designing algorithms for extracting the vehicle license plate from captured image and subsequently identifying the characters of the vehicle license plate. The proposed system has been implemented by using Vision Assistant 12 and LabVIEW 12. The performance of this system has been investigated for about 546 vehicles. The similar system has been executed in Matlab 7.0. The results obtained by both the software prove the efficacy of real-time laboratory prototype.

Keywords: Vehicle license plate (VLP) extraction, optical character recognition, Hough transform, pattern recognition.

Introduction

License plate recognition (LPR) is an image processing technology which is used to distinguish the vehicles through their license plates. The application is gaining popularity in traffic monitoring and security installations. Much research has been done for license plates of Korea, China, Australia, United States and other countries. The system is designed in such a way that license plates are recognized from the front as well as the rear view of the vehicle. The input to the system is the image taken through the digital camera consisting of the license plate and the output is the recognized characters of the license plate. The LPR system consists of four main modules as given below: (a) Image acquisition (b) License plate extraction (c) License plate segmentation (d) License plate recognition.

The first task is to acquire the image. The second task is to extract the region containing license plate. The third task is to segment the characters, letters and numerals (usually consisting of 10 letters in case of Indian license plates). The last task is to identify or recognize the segmented characters using optical character recognition (OCR).

Published papers in the domain of real-time VLP recognition are very scarce. However, there are several existing systems for extraction of license plates of different states. An exhaustive coverage would make the review too voluminous. Therefore, an attempt has been made to give an exposition to state of art in VLP recognition.

Naito[1] et. al. and Salgado[2] et. al. used a sensing system, which comprises of two CCDs (charge coupled devices) and a prism so that the an incident ray can be split into two lights with dissimilar intensities. The main characteristic of this sensing scheme is that it covers broad illumination conditions ranging from twilight to noon under sunshine. This system is also able to capture images of capable of capturing images of speeding vehicles without obscuring.

Laxmi[3, 4, 5] et. al. used extended Hough transform and Canny edge detection technique application on a predefined range for extraction of the license plate. Then neural network, fuzzy logic and SVM had been used to classify the wavelet coefficients obtained from different wavelets.

Al-Ghaili[6] et. al. and Sarfraz[7] et. al. used vertical edge detection algorithm (VEDA) to detect the license plate. Tahir[8] et. al. proposed vertical edge matching technique for extraction of the plate. Then vertical and horizontal scanning method has been used for character segmentation and template matching technique for recognition.

Guo[9] et. al. proposed symmetric mask-based discrete wavelet transform (SMDWT) algorithm combined with hierarchical adaboost (HA) technique for off-line training to detect multiple license plates in high resolution applications. Finally, structural/directional features with the naive bayes are applied for character recognition.

Ashtari[10] et. al. presented the method based on template-matching technique by analyzing the target color pixels for detection of license plate. Then a hybrid classifier is used for recognition of the license plate characters.

Villegas[11] et. al. proposed rectangular perimeter detection and the finding of a pattern by pattern matching for detection of the plate. Then the characters are extracted from the plate by means of horizontal and vertical projections. Finally, a fuzzy neural network has been used to recognize the license plate.

Zhou[12] et.al. presented an algorithm based on the combination of feature appearance and symmetry is proposed to locate the vehicle logo in a digital image.

Ozbay[13] et. al. used smearing algorithm to find the plate region. With this algorithm, the image is processed by vertical and horizontal scan-lines. If the number of white pixels is more than the desired threshold or lesser than the other threshold, white pixels are changed to black. After smearing, a morphological operation, dilation is implemented to the image for defining the plate location. However, there may be many candidate regions for the plate detection.

Kim[14] et. al. used two neural network-based filters, vertical and horizontal to examine small windows of the cross-sections of an image. These cross-sections contain sufficient information to identify a plate from the background.

Maglad[15] and Abulgasem[16] et. al. used radial basis function (RBF) neural network both for the detection as well as the recognition of Saudi Arabian and Libyan’s license plates respectively.

Jiang[17] et. al. proposed fused texture characteristics and color information based license plate location algorithm, where the license plate characteristics of structure and color are fused based on the fixed color collocation between background and the character of the license plate.

Zhou[18] et. al. used principal visual word (PVW) along with discovery and local feature matching. As the characters in different license plates are of same type, they used the bag-of words (BoW) model which can be used in partial-duplicate image search. For a new image, the license plates are located by corresponding local features with PVW.

Al-Audah[19] et. al. used system which relies on a series of filtering operation using morphological image processing. Then LabVIEW’s built in OCR tool had been used for character recognition. Nagare[20] used morphological operation to extract the plate and then uses neural network for character recognition for Indian license plates.

Kang[21] presented dynamic programming (DP) optimization for the functionality of the distribution of the character intervals, the character alignment. The threshold difference had been used to extract the character blobs. Li[22] proposed a geometric framework which detects the rectangular shape. The algorithm is based on three issues namely outer outliers, open shape and fragmentation. Finally, it provides the measure of rectangular shape by integrating the imbalanced points.

Du[23] et. al. and Anagnostopoulos[24] et. al. presented a comprehensive review of the various existing techniques for LPR. They compared different LPR techniques as per the features used in each step of the various algorithm, and then compared their advantages, drawback, recognition accuracy and processing speed.

In a nutshell, the research related to real-time VLP recognition, particularly for Indian subcontinent has not been done. The paper is organized as follows. Section II discusses the proposed system. Section III evaluates the methods used in this research work to extract the license plate. Section IV deals with the character recognition of the extracted license plate. Section V deals with the discussion and the results obtained and last section consists of conclusion of the research work. This paper embarks on implementation of a laboratory prototype of real-time VLP recognition system, as corroborated in the subsequent section.

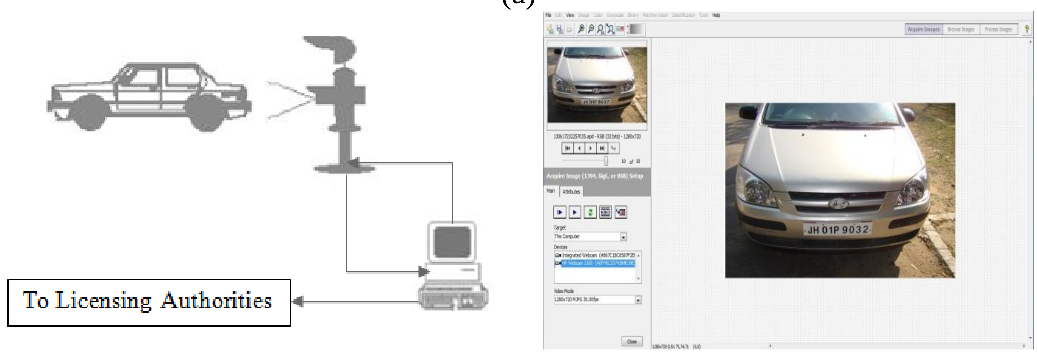
Laboratory Prototype System

A laboratory prototype requires set up for image acquisition, processing and subsequent VLP recognition in real-time. Keeping these requirements in view the designed system consists of a HP HD 3100 webcam connected to a computer having LabVIEW and MATLAB software for processing of images. The system can be installed at a parking or entrance of a restricted area or junction.

The purpose of using a webcam is to reduce the cost of the overall system. Since the range of the webcam is 2 to 10 feet with frame rate 30 frames per second (fps), so it works well for the purpose. The webcam is of 5MP and the image resolution is 1280×720. The different steps for the laboratory prototype system developed for the research work are:

Step 1: Capture the image by webcam.
Step 2: Transfer the image to the computer.
Step 3: Process image either in Matlab or LabVIEW.
Step 4: Recognition of VLP characters.
Step 5: Both the vehicle image and VLP characters are stored in the database.

(a)



(b) (c)

Figure 1: (a) Schematic diagram of LPR system (b) Image taken from webcam using Vision Assistant

Figure 1(a) shows the schematic diagram of a LPR system. Figure 1(b) shows the image taken from webcam using Vision Assistant. The computer consists of Intel core2duo with 2.1 GHz processor and 2 GB RAM with Microsoft Windows XP with SP2 as operating system.

The images have also been acquired from 8 MP Sony DSC-H7 digital camera from different places at different time using the same laboratory prototype for enhancing the database and checking for its efficacy. The images taken from camera are of size 3264×2448.

Methods Used For Extracting License Plate

The different methods used to extract the license plates in the research work are given in the subsequent sections.

Extraction of VLP Using Hough Transform

An acquired image is read in RGB format in Matlab and converted into grayscale in order to lessen the computational time for execution. The image is then cropped to concentrate on an area of interest which is mostly at the middle of the image. To overcome the complexity of the problem, the size of the original image is reduced as much as possible.

The system uses Canny edge detector to identify the edges of image of license plate, because it gives satisfactory results for the images with dirt also [3]. The horizontal and vertical edges forming license plate in the image are four straight lines in the image. These lines are obtained by extended Hough transform (EHT) as in [3].

The steps involved in the extraction of VLP are given in the algorithm below.

Step 1: Read the RGB image.

Step 2: Convert the color image into grayscale image.

Step 3: Crop the image to reduce the region of interest.

Step 4: Apply canny edge detection.

Step 5: Apply extended Hough Transform (EHT) to detect lines.

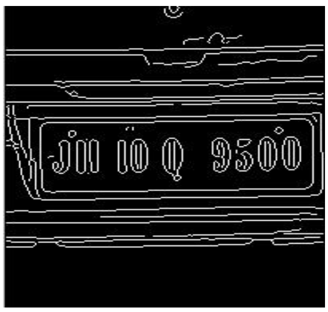
Step 6: If parallel lines are not in predefined range, discard the lines. Else, go to step 7.

Step 7: Determine the horizontal and vertical line segments.

Step 8: Crop the image where the horizontal and vertical lines intersect to extract the license plate.

Step 9: Input license plate to optical character recognition (OCR) tool for character recognition.

Figure 2(a) below shows cropping and application of Canny edge detector to the original image.



(a)

(b)

(c)

Figure 2: (a) Cropping of image and applying Canny edge detector (b) Applying Hough transform (c) Extracted license plate

Hough transform is a time consuming method for large images [3]. The main advantage of Hough transform is that, it is able to isolate features within the image and is unaffected by the image noise. The classical Hough transform dealt with the identification of lines in the image, but later the Hough transform have been extended to find the positions of arbitrary shapes. There are two types of Hough transform: standard Hough transform (SHT) and extended Hough transform (EHT). In SHT, this transform is applied on the whole image. In EHT, the transform is applied within the specified region. EHT has less execution time than that of SHT. In this work, EHT is used to detect the edges of license plate.

The Hough lines are obtained by calculating the Hough peaks of the image. The Hough lines are shown in the Figure 2(b). After the horizontal and vertical edges are obtained, license plate is detected by searching these Hough lines as shown in the Figure 2(c) above.

Extraction of VLP using Vision Assistant

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) 12, a graphical programming language software is used for data acquisition and instrument control. The advantages of using LabVIEW are to provide extensive support for accessing instrumentation hardware. It includes a compiler producing native code for the CPU platform for enhanced performance. It gives the information about real time controllers. Fully object oriented character of LabVIEW code allows code reuse without any modifications. Since it is expensive, low cost student edition is also available for learning process.

There are different steps to be followed in LabVIEW (Vision Assistant 12) for the development of script. Two different scripts have been developed for this purpose.

Vision Assistant Script 1

In the first method, using color matching technique the license plate of vehicle is located. The steps involved in the technique are given by the algorithm.

Step 1: Read the RGB image.

Step 2: Resample the image.

Step 3: Search for color template in RGB image.

Step 4: Compare the color content of template and the region of interest in the image.

Step 5: If the matching score is below 250, discard the image. Else, go to step 5.

Step 6: Extract the green plane of the image.

Step 7: Detect edges in the region of interest.

Step 8: Extract license plate.

Step 9: Input the license plate to optical character recognition (OCR) tool for character recognition.

The above steps are discussed below.

a. Original image

The image is obtained through 8.0 MP SONY DSC H7 digital camera and HP HD 3100 webcam in different places and at different time. The image is a 32-bit colored image.

b. Geometry
The image has been resampled to the size of 960×720 to reduce the processing time.

c. Color pattern matching
The image obtained is passed through this block, which contains a template image. It detects the coordinates of centroid of license plate in that image i.e., center X and center Y. It also gives the quality of match of candidate license plate with the template image i.e., score and angle θ with respect to the horizontal axis. The result of this is shown in Figure 3(a).

The angle is given by equation (1) as:

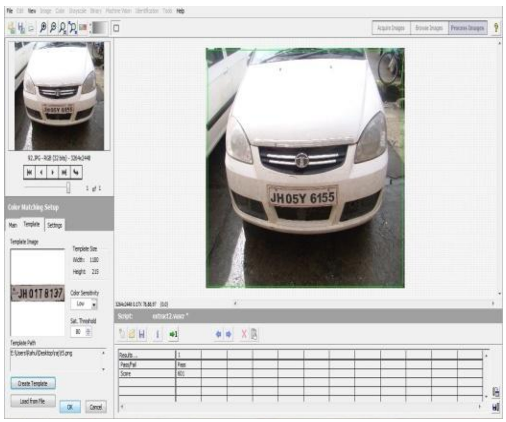
$$\theta = \tan^{-1}\frac{y}{x} \quad (1)$$

where, θ is angle and x & y are coordinates of centroid of license plate.

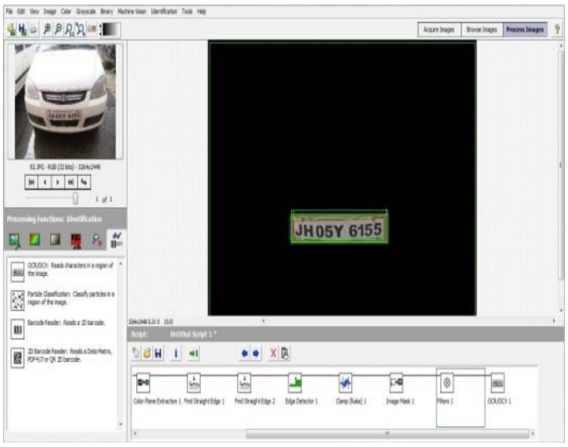
d. Color matching
The image obtained through color pattern matching is passed again through a template region where it compares the source image and it tells that whether the image is pass or fail. Hence it also gives the information about the score. If the score is above 250, then the image is passed otherwise it will give result as fail. If the score is 1000, it is perfect score. The above procedure is shown below with the help of Figure 3(b).

e. Extracting color planes from the image
Since the color image is redundant, 32-bit colored image is extracted to make it an 8-bit grayscale image. In this script, the green plane is selected from the image.

(a)



(b)



(c)

Figure 3: (a) Pattern matching of image (b) Color pattern matching of image (c) Image masking and low pass filter used in image

f. Brightness, contrast, gamma adjustment

Brightness of an image is defined as the perceived luminance and depends on the luminance of the surrounding [25]. Brightness is expressed in gray levels, centered at 128. Higher values (up to 255) result in a brighter image, lower values (down to 0) results in a darker image. In this algorithm, the level of brightness is 140.

Contrast of image is expressed as an angle. A value of 45° specifies no change in contrast. Higher value (up to 89°) increase contrast and lower value (up to 1°) decrease contrast. In this method, the level of contrast is 50.00°.

Gamma value is used for gamma correction. It is a process to correct the power law response phenomena [26] and given by equation (2) as:

$$s = cr^{\gamma} \qquad (2)$$

where, r is input intensity level, s is output intensity level and c is positive constant.

The higher the gamma coefficient, stronger is the intensity correction. In this method, the value of the gamma is 1.04.

g. Locating the straight edges

By doing different adjustments in image as in previous step, the straight lines are now located in the region of interest in the image by plotting the edge strength profile of the image.

A simple edge detector using gradient magnitude computes gradient vector at each pixel by convolving image with horizontal and vertical derivative filters. It also computes gradient magnitude at each pixel. For finding edge strength and direction at location (x, y) of an image, f, is the gradient and is defined as the vector as in equation (3):

$$\nabla f \equiv \mathrm{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^T \qquad (3)$$

The magnitude is given by equation (4):

$$\mathrm{mag}(\nabla f) = \left(g_x^2 + g_y^2\right)^{0.5} \tag{4}$$

and the direction of gradient vector is given by equation (5):

$$\alpha(x, y) = \tan^{-1}\left[\frac{g_y}{g_x}\right] \tag{5}$$

If magnitude at a pixel is more than a threshold, there is a possible edge point. Edge detection involves 3 steps i.e., noise smoothing, edge enhancement and edge localization [26].

h. Clamp of image

After locating the straight lines of the images, current distance (in pixels) separating the object edges of the images is also calculated and edge strength profile of that image is plotted. As a result, the width of the license plate could be known.

i. Image masking

An image mask isolates the part of the image for processing. By doing different adjustments, straight lines are located in the region of interest in the image by obtaining the edge strength profile. An image mask consists of an 8-bit binary image and is of same size or smaller than inspection image.

Pixels of image determine whether the corresponding pixels of image are to be processed or not. If a pixel of image mask has value other than zero, the corresponding pixel in the inspection image is processed. If the pixel of the image mask is zero, then the corresponding pixel of the inspection image is not worked on. Pixels in the source image are worked on if the corresponding pixels of the image mask have higher values of zero. Mask of the image is affected if the pixel value of the image is inverted [27].

In this method, by applying the image mask covering the whole of license plate and therefore extracting the masked region by creating the region of interest (ROI) for the licese plate so that it can go for further process in order to have good efficiency.

j. Filters

Low pass filters also known as smoothing filters are used to discard high spatial frequency noise from a digital image. There are several common approaches to remove this noise. In the first approach, if multiple copies of an image are found from the source, then it is possible to add the values of each pixel from each image and calculate the average. If the image is obtained from a moving source, it is not possible. There are also size restrictions. In another approach, if such averaging is not possible, or if it is insufficient, some sort of low pass spatial filtering may be needed [27]. The result of the above procedure is given below in Figure 3(c).

Vision Assistant Script 2

In this method pattern matching technique is used to recognize the license plates. Pattern matching is a family of tools for finding similar objects in different sources. In image processing, the pattern matching is used for locating a small image (called reference image) as in Figure 4(a), in a bigger one (called search image). The simple way is to move the small image in the bigger image and measure a similarity in each position. The position with highest value of the similarity is accepted as result.

For that purpose, at each position of the reference image in the search image, a similarity value, e.g., the cross-correlation coefficient $k_{R,S}$ of the gray levels is calculated in equation (6):

$$k_{R,S}(\Delta r, \Delta c) = \frac{\sum_{r_R, c_R}[g_R(r_R, c_R) - \bar{g}_R] \cdot [g_S(r_R + \Delta r, c_R + \Delta c) - \bar{g}_S]}{\sqrt{\sum_{r_R, c_R}[g_R(r_R, c_R) - \bar{g}_R]^2 \cdot \sum_{r_R, c_R}[g_S(r_R + \Delta r, c_R + \Delta c) - \bar{g}_S]^2}} \quad (6)$$

In the above equation, $\bar{g}_R$ and $\bar{g}_S$ denote the arithmetic mean gray level in the reference image and the part of the search image covered by the reference image respectively. All sums are to be taken over all pixels of the reference image [28].

The position of the point corresponding to the reference point is given by the position of the maximum of the similarity measure. In case the cross-correlation coefficient is used, the threshold can be chosen rather easily (e.g., 0.8) because the coefficient is bounded by 0 and 1 [28]. For this method the value has been normalized to the range from 0 to 1000.

There are different steps in LabVIEW (Vision Assistant 12) for the development of script as are given by the algorithm.

Step 1: Read the RGB image.

Step 2: Resample the image.

Step 3: Extract the green plane of the image.

Step 4: Match the template using pattern matching in the region of interest in the image.

Step 5: If the matching score is below 250, discard the image. Else, go to step 5.

Step 6: Detect left, top, right and bottom edges using Hough edge in the region of interest.

Step 7: Extract license plate.

Step 8: Input the license plate to optical character recognition (OCR) tool for character recognition.

The above steps are discussed below.

a. Original image

The 32-bit colored image is input in the first step.

b. Geometry

In this step, the image is resized to 960×720. The reduced pixels will decrease the processing time of the algorithm.

c. Extracting color planes from the image

The color image of 32-bit is converted into an 8-bit grayscale image. In this script, the green plane is selected from the image. The procedure is given below in Figure 4(b).

d. Pattern matching

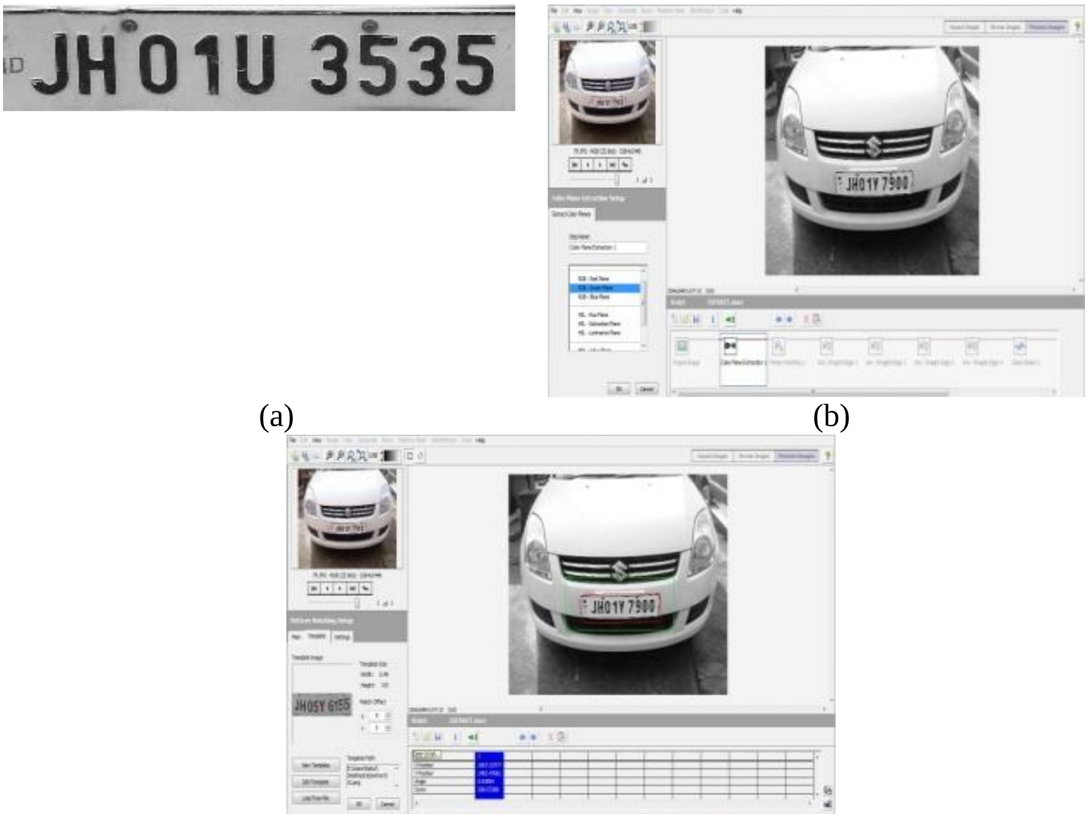
In this step, a template is selected and a region of interest has been set in the 8-bit grayscale image. Figure 4(a) shows the template image and is of size 304×70. The block looks for the similar model in the target region. Thus, it detects the license plate in the region of interest. It gives the results of center X, center Y, score and angle. Center X gives X-coordinate of each object that matches the template. Center Y gives Y-coordinate of each object that matches the template. Score indicates the score of each valid match [28]. If the score is above 250, then the image will pass for further processing. If the score is 1000, it indicates perfect match. Figure 4(c) gives the above procedure.

e. Locating the edges

The next step is to locate the straight lines in region of interest in the image and plot the edge strength profile in the image. The left, top, right and bottom edges have been detected using Hough edge. Like in the previous script after locating the straight lines of the images, current distance (in pixels) separating the object edges of the images is calculated and edge strength profile of that image is plotted.

f. Image masking and Filters

Like previous script the image has been masked and thus the license plate is extracted. For smoothing the noises, low pass filters has been used.



(a) (b)

(c)

Figure 4: (a) Template image (b) Color Extraction (c) Pattern matching of image

Character Recognition

In the case of optical character recognition (OCR) (also regarded as off-line), printed or handwritten text will be represented by a bit-mapped image typically from a scanner. The process of OCR requires segmentation and a preprocessing stage [29]. The algorithm for character recognition in OCR tool is:

Step 1: Train the OCR tool with a stored set of characters.
Step 2: VLP image is input to the OCR.
Step 3: Draw the region of interest.
Step 4: Set the threshold.
Step 5: Read the text on the VLP.

OCR works by first preprocessing the digital image into its smallest part component with layout analysis to find the text blocks/ sentence/ line blocks, word blocks and the character blocks. The features are then extracted by drawing out the relevant information. Other features such as lines, graphics, and photographs are recognized and discarded [30].

The characters blocks are further broken down into components, pattern are recognized and compared to OCR engines containing large dictionary of characters from various fonts and languages [29].

The character recognition requires abundant supply of test data to train the OCR.

The OCR tool has been trained with stored set of characters. The training was done off-line and for specific size, font and spacing of the characters in the license plate. Figure 5 displays the OCR training interface.

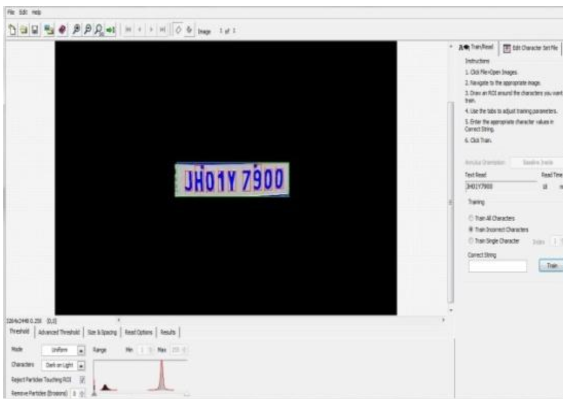


Figure 5: OCR training interface

Results and Discussion

In the extraction stage, a test is performed and result is shown for the vehicle. The extraction of the license plate is based on four steps: edge detection, Hough lines detection, selection of edges of license plate and cropping of selected region depending upon the corner points of the license plate.

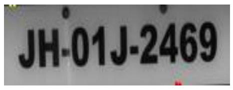
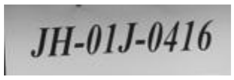
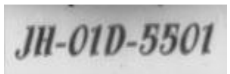
For the proposed system, same assumptions are taken as in [3]. So, the efficiency of the license plate extraction depends upon efficient image capturing.

The recognition stage is the final step of license plate recognition system. Recognition of license plate is done by using Matlab7.0 and the segmentation of characters is done by using LabVIEW and Vision Assistant 12. The results of the license plate recognition by using two different softwares have been discussed below.

In Matlab programming is done by programming language in which whole license plate is extracted, whereas in LabVIEW programming is done by using function blocks in which OCR training interface is used. Matlab requires large processing time as compared to LabVIEW for large processes.

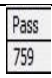
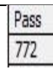
The images which are simulated in Matlab7.0 are shown below in Table I.

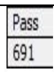
Table 1: Results obtained from Matlab

S. No.	Input Image	Extracted License Plate
1.		
2.		
3.		

The results of some of the images are shown below in Table II with correct no. and its number shown using Vision Assistant.

Table 2: Results obtained from Vision Assistant

S. No.	Input Image	Correct No.	Score	Number read by Vision Assistant
1.		JH01H8640		
2.		JH01Y9278		

3.		JH01R4729		

From the previous results analysis, it can be concluded that the plate extraction using Hough lines gives satisfactory results and the vehicle license plate is extracted. In this proposed method, LabVIEW is used for the segmentation of characters.
Table III below gives a brief comparison of results obtained from the three methods. A total of 546 images have been tested. The three methods have been tested for a set of 163, 197 and 186 different images each respectively.

Table 3: Comparisons of three methods

Methods	License Plate Extraction Method	No. of images	Correct VLP Extraction	Correct Recognition	No. of Failures	Recognition Rate (%)	Plate format	Average Processing Time (msec)
Hough Transform	Edge detection and Hough lines	163	159	139	24	85.2	Indian	192.2
Vision Assistant Script 1	Template matching	197	193	167	20	84.7	Indian	131.6
Vision Assistant Script 2	Template matching	186	182	145	21	77.9	Indian	71.3

In the process of evaluation of the research work, some parameters such as brightness, contrast, taking the image from certain distance has been optimized so that the character segmentation is obtained in the OCR training.

Conclusion
The proposed real-time LPR is simple both in terms of overall components used and its efficacy for dealing with possible application for traffic management and licensing authorities. Though the system has been tested for light motor vehicles (LMV), the future research is target for all types of motor vehicles. The computational efficiency for real-time as well as success rate of recognition seem to be promising for pragmatic application on a mass scale.

References

[1] Naito, T., Tsukuda, T., Yamada, K., Kozuka, K., and Yamato S., 1999, “License plate Recognition System for inclined plates outdoors,”

Proceeding International Conference on Information Intelligence and Systems, U.S., pp. 304-312.
[2] Salagado, L., Menendez, J. M., Rendon, E., and Garcia, N., 1999, "Automatic car plate detection and recognition through intelligent vision engineering," Proceedings IEEE 33rd Annual International Carnahan Conference on Security Technology, Spain, pp. 71-76.
[3] Laxmi, V., Mohanta, D. K., and Karan, B. M., 2013, "Wavelet based intelligent techniques for automatic identification of motor vehicles: A comparative study," International Journal of Vehicle Autonomous Systems, 11(1), pp. 42-61.
[4] Laxmi, V., Mohanta, D. K., and Karan, B. M., 2011, "Comparison of different wavelets for automatic identification of vehicle license plate," IET Intelligent Transport Systems, 5(4), pp. 231–240.
[5] Laxmi, V., and Karan, B. M., 2010, "Vehicle identification automation using wavelet-probabilistic neural network combined approach," International Journal of Signal and Image Processing, 1(2), pp. 62-68.
[6] Al-Ghaili, A. M., Mashohor, S., Ramli, A.R., and Osmail,A., 2013, "Vertical-edge based car-license-plate detection method," IEEE Transactions on Vehicular Technology, 62(1), pp. 26-38.
[7] Sarfraz, M., Ahmed, M. J., and Ghazi, S. A., 2003, "Saudi Arabian license plate recognition system," International Conference on Geometric Modeling and Graphics, England, pp. 36-41.
[8] Tahir, A., Habib, H. A., and Khan, M. F., 2010, "License plate recognition algorithm for Pakistani license plates," Canadian Journal on Image Processing and Computer Vision, 1(2), pp. 30-36.
[9] Guo, J., Liu, Y., and Hsia, C., 2012, "Multiple license plates recognition system," International Conference on System Science and Engineering. Liaoning, pp. 120-125.
[10] Ashtari, A. H., Nordin, M. J., and Fathy, M., 2014, "An Iranian license plate recognition system based on color features," IEEE Transactions on Intelligent Transportation Systems, 15(4), pp. 1690-1705.
[11] Villegas, O. O. V., Balderrama, D. G., Domínguez, H. J. O., and Sánchez, V. G. C., 2009, "License plate recognition using a Novel Fuzzy Multilayer neural network," International Journal of Computers, 3(1), pp. 31-40.
[12] Zhou, K., Varadarajan, K. M., Vincze, M., and Liu, F., 2012, "Hybridization of Appearance and Symmetry for Vehicle-Logo Localization," 15th International IEEE Conference on Intelligent Transportation Systems, U.S., pp. 1396-1401.
[13] Ozbay, S., and Ercelebi, E., 2007, "Automatic vehicle identification by plate recognition," World Academy of Science, Engineering and Technology, 9, pp. 778-781.
[14] Kim, K. K., Kim, K. I., Kim, J.B., and Kim, H. J., 2000, "Learning based approach for license plate recognition," Proceedings of IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing, NSW, pp. 614-623.

[15] Maglad, K. W., 2012, “A vehicle license plate detection and recognition system,” Journal of Computer Science, 8(3), pp. 310-315.
[16] Abulgasem, N. A., Mohamad, D., and Hasim, S. Z. M., 2011, “Automatic license plate detection and recognition using radial basis function neural network,” International Journal of Computer Vision and Applications, 1(1), pp. 15-23.
[17] Jiang, X., Huang, Y., and Li, G., 2012, “Fusion of texture features and color information of the license plate location algorithm,” International Conference on Wavelet Analysis and Pattern Recognition, China, pp. 15-19.
[18] Zhou, W., Li, H., Lu, Y. and Tian, Q., 2012, “Principal visual word discovery for automatic license plate detection,” IEEE Transactions on Image Processing, 21(9), pp. 4269-4279.
[19] Al-Audah Y. K., Al-Juraifani A. K. and Deriche M. A., 2012, “A real-time license plate recognition system for Saudi Arabia using LabVIEW,” IEEE International Conference on Image Processing Theory, Tools and Applications, Turkey, pp. 160-164.
[20] Nagare, A. P., 2011, “License plate character recognition system using neural network,” International Journal of Computer Applications, 25(10), pp. 36-39.
[21] Kang, D. J., 2009, “Dynamic programming-based method for extraction of license plate numbers of speeding vehicles on the highway,” International Journal of Automotive Technology, 10(2), pp. 205-210.
[22] Li, Q., 2014, “A geometric framework for rectangular shape detection,” IEEE Transactions on Image Processing, 23(9), pp. 4139-4149.
[23] Du, S., Ibrahim, M., Shehata, M., and Badawy, W., 2013, “Automatic license plate recognition (ALPR): A State-of-the-Art review,” IEEE Transactions on Circuits and Systems for Video Technology, 23(2), pp. 311-325.
[24] Anagnostopoulos, C. E., Anagnostopoulos, I. E., Psoroulas, I. D., Loumos, V., and Kayafas, E., 2008, “License plate recognition from still images and video sequences: A survey,” IEEE Transactions on Intelligent Transportation Systems, 9(3), pp. 377-391.
[25] Jain A. K., 2011, Fundamentals of Digital Image Processing, 1st edition, pp. 51.
[26] Gonzalez R. C., Woods R., and Eddins S. L., 2012, A Textbook for Digital Image Processing using MATLAB, 3rd edition, pp. 110-111, pp. 704-708.
[27] http://www.ni.com/pdf/manuals accessed April 2015.
[28] Kropatsh, W. G., and Bischof H., 2001, Digital Image Analysis Selected Techniques and Applications, 1st edition, pp. 395-396.
[29] Pandya, A. S., and Macy, R. B., 1996, Pattern Recognition with Neural Networks in C++ ,1st edition, pp. 19.
[30] Chan, K., and Yeung, D., 2000, “Mathematical expression recognition: A survey,” International Journal on Documentation Analysis and Recognition, 3(1), pp. 3-15.