Car License Plate Recognition Using VEDA

Anish A
II YR M.E-COMMUNICATION SYSTEM
Maharaja Prithvi Engineering College, Avinashi, Coimbatore , India
anielphin84@gmail.com

Ms.P.Nithya
Assistant Professor, Department of Electronics and communication Engineering , Maharaja Prithvi Engineering College,Avinashi,Coimbatore , India
nithyapk1289@gmail.com

Abstract— **License Plate Recognition System is a challenging problem in the field of machine vision and automation with various applications. Detection is a part of License Plate Recognition technique, used to find out a vehicle by its number plate. In this project we propose a noise insensitive license plate detection method. The detection technique is used for discriminating between corrupted and uncorrupted image pixels. The corrupted pixels are restored using weighted trimmed median filter. Our proposed filter is very effective in removing impulse noise without destroying the useful information contained in the image data for efficient extraction. After filtering, an unwanted-line elimination algorithm (ULEA) is presented to enhance the image, and then, the VEDA is applied. The proposed algorithm focuses on detecting the characters of two rows license plate image. We present a new and fast vertical edge detection algorithm (VEDA) which is based on the contrast between the gray scale values. Before the detection algorithm is applied, the License plate must be binarized by using adaptive threshold, and apply unwanted lines elimination algorithm (ULEA). At last, an LP is detected. The results show accurate edge detection performance and demonstrated the great efficiency of using VEDA in order to highlight license plate details.**

Keywords- **car-license-plate detection (CLPD), sobel operator, vertical edge detection algorithm (VEDA).**

I. INTRODUCTION

The car-license-plate (CLP) recognition system is an image processing technology used to identify vehicles by capturing their CLPs. The CLP recognition technology is known as automatic number-plate recognition, automatic vehicle identification, CLP recognition, or optical character recognition for cars. The CLP detection and recognition system (CLPDRS) became an important area of research due to its various applications, such as the payment of parking fees, highway toll fees, traffic data collection, and crime prevention.

Usually, a CLPDRS consists of three parts: license-plate (LP) detection (LPD), character segmentation, and character recognition. Among these, LPD is the most important part in the system because it affects the system's accuracy.

There are many issues that should be resolved to create a successful and fast CLP detection system (CLPDS), e.g., poor image quality, plate sizes and designs, processing time, and background details and complexity. The need for car identification is increasing for many reasons such as crime prevention, vehicle access control, and border control. To identify a car, features such as model, color, and LP number can be used.

In vehicle tracking systems, cameras are used and installed in front of police cars to identify those vehicles. Usually, numerous vehicle tracking and pursue systems use outstanding cameras, and this leads to cost increment of the system in both hardware and software. Since many methods have been proposed in various intelligent transportation system applications, the CLPDRS is usually based on an image acquired at 640 × 480 resolution. An enhancement of CLPD method performance such as reduction of computation time and algorithm complexity, or even the build of the LP recognition system with lower cost of its hardware devices, will make it more practical and usable than before. This paper proposed a method for CLPD, in which a web camera with 352× 288 resolution is used instead of a more sophisticated web camera.

Web camera is used to capture the images, and an offline process is performed to detect the plate from the whole scene image.

Vertical edge extraction and detection is an important step in the CLPDRS because it affects the system's accuracy and computation time. Hence, a new vertical edge detection algorithm (VEDA) is proposed here to reduce the computation time of the whole CLPD method.

A vehicular ad hoc network (VANET) uses moving cars as nodes in a network to create a mobile network. A VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 meters of each other to connect and, in turn, create a network with a wide range. As cars fall out of the signal range and drop out of the network, other cars can join in, connecting vehicles to one another so that a mobile Internet is created. It is estimated that the first systems that will integrate this technology are police and fire vehicles to communicate with each other for safety purposes.

Intelligent vehicular ad hoc networks (In VANETs) use Wi-Fi and Wi-MAX for easy and effective communication between vehicles with dynamic mobility. Effective measures such as media communication between vehicles can be enabled as well methods to track automotive vehicles. In VANET is not foreseen to replace current mobile communication standards.

Automotive vehicular information can be viewed on electronic maps using the Internet or specialized software. The advantage of WiFi based navigation system function is that it can effectively locate a vehicle which is inside big campuses like universities, airports, and tunnels. In VANET can be used as part of automotive electronics, which has to identify an optimally minimal path for navigation with minimal traffic intensity. The system can also be used as a city guide to locate and identify landmarks in a new city.

Communication capabilities in vehicles are the basis of an envisioned In VANET or intelligent transportation systems (ITS). Vehicles are enabled to communicate among themselves (vehicle-to-vehicle, V2V) and via roadside access points (vehicle-to-roadside, V2R). Vehicular communication is expected to contribute to safer and more efficient roads by providing timely information to drivers, and also to make travel more convenient. The integration of V2V and V2R communication is beneficial because V2R provides better service sparse networks and long distance communication, whereas V2V enables direct communication for small to medium distances/areas and at locations where roadside access points are not available.

Providing vehicle–vehicle and vehicle–roadside communication can considerably improve traffic safety and comfort of driving and traveling. For communication in vehicular ad hoc networks, position-based routing has emerged as a promising candidate. For Internet access, Mobile IPv6 is a widely accepted solution to provide session continuity and reach ability to the Internet for mobile nodes. While integrated solutions for usage of Mobile IPv6 in (non-vehicular) mobile ad hoc networks exist, a solution has been proposed that, built upon a Mobile IPv6 proxy-based architecture, selects the optimal communication mode (direct in-vehicle, vehicle–vehicle, and vehicle–roadside communication) and provides dynamic switching between vehicle–vehicle and vehicle–roadside communication mode during a communication session in case that more than one communication mode is simultaneously available.

Currently there is ongoing research in the field of In VANETs for several scenarios. The main interest is in applications for traffic scenarios, mobile phone systems, sensor networks and future combat systems. Recent research has focused on topology related problems such as range optimization, routing mechanisms, or address systems, as well as security issues like traceability or encryption. In addition, there are very specific research interests such as the effects of directional antennas for In VANETs and minimal power consumption for sensor networks. Most of this research aims either at a general approach to wireless networks in a broad setting or focus on an extremely specific issue.

II. APPLICATIONS

Most of the concerns of interest to mobile ad-hoc networks (MANETs) are of interest in VANETs, but the details differ. Rather than moving at random, vehicles tend to move in an organized fashion. The interactions with roadside equipment can likewise be characterized fairly accurately. And finally, most vehicles are restricted in their range of motion, for example by being constrained to follow a paved highway.

In addition, in 2006 the term MANET mostly described an academic area of research, and the term VANET an application.

Such a network might poses safety concerns (for example, one cannot safely type an email while driving). GPS and navigation systems might benefit, as they could be integrated with traffic reports to provide the fastest route to work. It was also promoted for free, VoIP services such as Google Talk or Skype between employees, lowering telecommunications costs.

Security issues:

VANET security should satisfy four goals, it should ensure that the information received is correct (information authenticity), the source is who he claims to be (message integrity and source authentication), the node sending the message cannot be identified and tracked (privacy) and the system is robust. Several attacks can be identified and these can be generalized depending on the layer the attacker uses. At the physical and link layers the attacker can either disturb the system by jamming or overloading the channel with messages. Injecting false messages or rebroadcasting an old message is another possible attack. The attacker can also steal or tamper with a car system or destroy a RSU. At the network layer the attacker can inject false routing messages or overload the system with routing messages. The attacker can also compromise the privacy of drivers by revealing and tracking the positions of the nodes. The same attacks can be achieved from the application layer.

In the IEEE WAVE standard vehicles can change their IP addresses and use random MAC addresses to achieve security. Vehicles also keep the message exchange to a minimum at the start of the journey for some time so that the messages cannot be tied to the vehicle. A number of security algorithms have been developed in France Telecom R&D department. The security proposal provides security at the link layer for vehicle safety and commercial applications, higher layer security protocols can also be used to further enhance the security or provide end to end security in a multihop link. The proposal makes use of four types of certificates, two long terms and two short terms.

One long term and one short term certificates are used for ITS services while the others are for non ITS applications. Long term certificates are used for authentication while short term certificates are used for data transmission using public/private key cryptography. Safety messages are not encrypted as they are intended for broadcasting, but their validity must be checked; Therefore a source signs a message and sends it without encryption with its certificate; other nodes receiving the message validate it using the certificate and signature and may forward it without modification if it is a valid message. Non ITS data can rely on higher layer protocols to provide end to end security especially over a multihop link.

The proposal suggests the use of a long term certificate, issued by a governmental authority (GA), and temporary certificates, issued by private authorities (PA), as well as pseudonyms to protect the privacy of the drivers. For commercial services, if the user is communicating directly with the RSU, its identity is validated via the long term certificate by the GA and then it is issued a temporary certificate and pseudonym by the PA to be able to use the service. For communications via hops the source signs the message using the long term certificate, forwarding vehicles verify the message and sign it using their own certificates and so on till it reaches the RSU.

III RELATED WORK

There are several LPD methods that have been used before, such as morphological operations, edge extraction, combination of gradient features, salience features, a neural network for color or grayscale classification, and vector quantization. Kim et al. proposed an LPD algorithm using both statistical features and LP templates. After the statistical features were used to select the regions of interest (ROIs), LP templates were applied to match the ROI.Matas and Zimmermann proposed an algorithm to detect LPs under various conditions. Their algorithm used character regions as basic units of LPs, which make the algorithm quite robust to view-point and illumination. An LPD algorithm using color edge and fuzzy disciplines has been proposed.
A vertical edge map has been used for LPD for many years. The given algorithms used a one-directional Sobel operator to extract the vertical edges. Nevertheless, some undesired details such as horizontal edges are kept in such vertical edge map.

An image enhancement and Sobel operator was used to extract the vertical edges of the car image. They used an algorithm to remove most of the background and noisy edges.Zhang et al. defined a new vertical gradient map to extract statistical features. The authors constructed two cascade classifiers based on statistical and Haar features to decrease the complexity of the system and to improve the detection rate.Baiet al. proposed an algorithm for LPD for monitoring the highway ticketing systems. Their algorithm presented a linear filter to smooth the image and to overcome the influence of light. The most common and earliest edge detection algorithms are those based on the gradient, such as the Sobel operator and the Roberts operator. Numerous previous methods have used the Sobel operator to extract the vertical edges in CLPDSs.

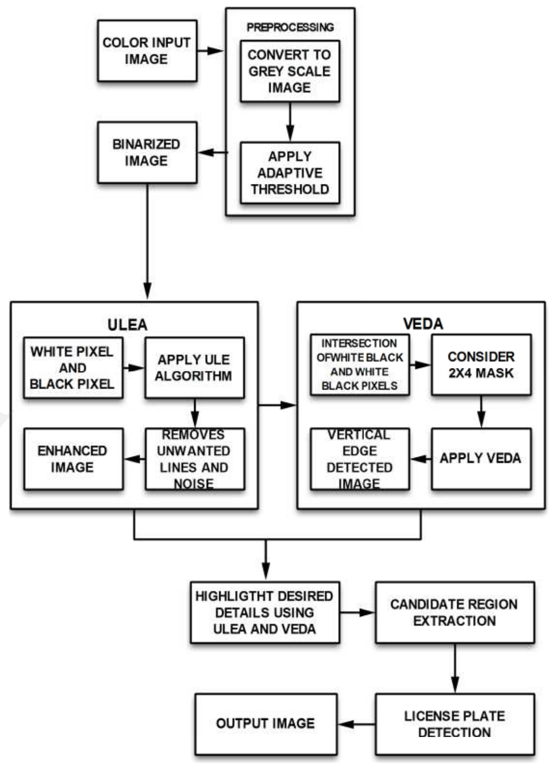

Fig.1 Flowchart of the proposed system

IV. PROPOSED SYSTEM FOR CAR LICENSE PLATE DETECTION

A. *Overview*

This paper has three contributions: The VEDA is proposed and used for detecting vertical edges; the proposed CLPD method processes low-quality images produced by a web cam-era, which has a resolution of 352 × 288 with 30 fps; and the computation time of the CLPD method is less than several methods. In this paper, the color input image is converted to a grayscale image, and then, adaptive thresholding (AT) is applied on the image to constitute the binarized image. Afte rthat, the ULEA is applied to remove noise and to enhance the binarized image. Next, the vertical edges are extracted by using the VEDA. The next process is to detect the LP; the plate details are highlighted based on the pixel value with the help of the VEDA output. Then, some statistical and logical operations are used to detect candidate regions and to search for the true candidate region. Finally, the true plate region is detected in the original image.

B. AT

After the color input image is converted to grayscale, an AT process is applied to constitute the binarized image. Bradley and Roth recently proposed real-time AT using the mean of a local window, where local mean is computed using an integral image. To get a good adaptive threshold, the method proposed is used.

[1] *Technique of AT***:** The AT technique used in this paper is just a simple extension of Bradley and Roth's and Wellner's methods. The idea in Wellner's algorithm is that the pixel is compared with an average of neighboring pixels.

Specifically, an approximate moving average of the last S pixels seen is calculated while traversing the image. If the value ofthe current pixel is T percent lower than the average, then it is set to black; otherwise, it is set to white. This technique is useful because comparing a pixel to the average of neighboring pixels will keep hard contrast lines and ignore soft gradient changes. The advantage of this technique is that only a single pass through the image is required. Wellner uses one eighth of the image width for the value of S and 0.15 for the value of T to yield the best results for a variety of images. The value of T might be a little bit modified from the proposed value by Wellner depending on the used images; whereas it should be in the range 0.1 <T< 0.2 in our method.

However, Wellner's algorithm depends on the scanning or-der of pixels. Since the neighborhood samples are not evenly distributed in all directions, the moving average process is not suitable to give a good representation for the neighboring pixels. Therefore, using the integral image has solved this problem.

[2] *AT Formulations:* The first step is to compute the integralimage. Initially, the summation of the pixel values for every column j th through all row values i will be computed using

$$\mathbf{sum}(i)\bigg|_{j^{\mathrm{th}}} = \sum_{x=0}^{i} g(x,y)\bigg|_{y=j^{\mathrm{th}}}$$

where g (x,y) represents the input values, and sum (i) |jth rep-resents all cumulative gray values of g (x,y) for the column j ththrough all rows of image I = 0, 1 ,... height.

Then, the integral image can then be computed for every pixel

$$\mathrm{IntgrlImg}(i,j) = \begin{cases} \mathrm{sum}(i), & \text{if } j = 0 \\ \mathrm{IntgrlImg}(i, j-1) + \mathrm{sum}(i), & \text{otherwise} \end{cases}$$

whereIntgrlImg(i,j) represents the integral image forpixel (i,j) .

The next step is to perform thresholding for each pixel. To do so, first, the intensity summation for each local window should be computed by using two subtraction operations and one addition operation as follows:

$$\begin{aligned} \mathbf{sum}_{\mathrm{window}} = & \left(\mathbf{IntgrlImg}\left(i + \frac{s}{2}, j + \frac{s}{2}\right)\right) \\ & - \left(\mathbf{IntgrlImg}\left(i + \frac{s}{2}, j - \frac{s}{2}\right)\right) \\ & - \left(\mathbf{IntgrlImg}\left(i - \frac{s}{2}, j + \frac{s}{2}\right)\right) \\ & + \left(\mathbf{IntgrlImg}\left(i - \frac{s}{2}, j - \frac{s}{2}\right)\right) \end{aligned}$$

Where $\mathrm{sum}_{\mathrm{window}}$ represents the summation of the intensities of the gray values for a specified local window, in which the currently binarized pixel is centering in. The boundaries of the window can be represented by

$$\left(i + \frac{s}{2}, j + \frac{s}{2}\right), \left(i + \frac{s}{2}, j - \frac{s}{2}\right), \left(i - \frac{s}{2}, j + \frac{s}{2}\right), \left(i - \frac{s}{2}, j - \frac{s}{2}\right)$$

and s represents the local window size/lengths for the computed IntgrlImg, whereas s = image width /8.Therefore, to compute the adaptive threshold value for the image, in which g (i,j) ∈ [0 , 255] is the intensity of the pixel located at (i,j) , threshold t (i,j) for each pixel has to be computed first as follows:

$$t(i,j) = (1 - T) \times \mathrm{sum}_{\mathrm{window}}$$

Where t(i,j) represents the threshold for each pixel at location(i,j) , and T is a constant, i.e., T = 0.15. This value is the optimal value for best thresholding performance for the whole images after testing on many images and is assessed visually.

The criterion in the following is applied on every pixel to output the threshold value of that pixel:

$$o(i,j) = \begin{cases} 0, & g(i,j) \times S^2 < t(i,j) \\ 255, & \text{otherwise} \end{cases}$$

where o (i,j) represents the adaptive threshold output value of pixel g (i,j) , and S^2represents the computed area of the local window for the selected region.

[3] *Effect of T Value on AT:* Here, the effect of the T value will be explained to evaluate the sensitivity of the image. Two examples will be given here: The first example has a lesser value of T than the pre given value and the other example has a higher value. In, if T = 0. 05 →t (i,j)= 0. 95 ×$\mathrm{sum}_{\mathrm{window}}$, then condition g (i,j) × S^2<t(i,j) will betrue in most cases. Therefore, adaptive threshold o(i,j)= 0and a lot of gray regions g (i,j) in the input image will appearing the binarized image as black regions. On the contrary, ifT = 0.29 →t (i,j)= 0. 71 × $\mathrm{sum}_{\mathrm{window}}$, then the condition will be false. That is, important details will disappear. After testing a lot of values, the most adequate value for T has been found as 0.15 for all of the input images.

As a result, the decrement of T below 0.15 will affect in constituting new black regions, and vice versa, the increment of T above 0.15 will affect in eliminating important details shows the input image, and the result is shown after AT is applied.

C. ULEA

Thresholding process in general produces many thin lines that do not belong to the LP region we can see that there are many long foreground lines and short random noise edges beside the LP region. These background and noise edges are unwanted lines. These lines may interfere in the LP location. Therefore, we have proposed an algorithm to eliminate them from the image. This step can be considered as a morphological operation and enhancement process. There are four cases in which unwanted lines can be formed. In the first case, the line is horizontal with an angle equal to 0˚as (−). In the second case, the line is vertical with an angle equal to 90˚as (|). In the third case, the line is inclined with an angle equal to 45˚as (/). In the fourth case, the line is inclined with an angle equal to 135˚as(\). Therefore, the ULEA has been proposed to eliminate these lines. In this step, while processing a binary image, the black pixel values are the background, and the white pixel values are the foreground. A 3 × 3 mask is used throughout all image pixels. Only black pixel values in the thresholded image are tested. To retain the small details of the LP, only the lines whose widths equal to 1 pixel are checked. Suppose that b(x,y) are the values for thresholded image. Once, the current pixel value located at the mask center is black, the eight-neighbor pixel values are tested. If two corresponding values are white together, then the current pixel is converted to a white value as a foreground pixel value (i.e., white pixel) shows the possible cases in which the current pixel is converted to foreground pixel. Each case of (a)–(d) represents two corresponding values at each time that the mask moves through the image shows the output after the ULEA is performed, whereby many unwanted lines are removed from the image. This kind of image is nearly ready for a better segmentation process.

D. VEDA

The advantage of the VEDA is to distinguish the plate-detail region, particularly the beginning and the end of each character. Therefore, the plate details will be easily detected, and the character recognition process will be done faster. After thresholding and ULEA processes, the image will only have black and white regions, and the VEDA is processing these regions. The idea of the VEDA concentrates on intersections of black–white and white–black. A2 × 4 mask is proposed for this process, as shown ,where x and y represent rows and columns of the image, respectively. The center pixel of the mask is located at points (0, 1) and (1, 1). By moving the mask from left to right, the black–white regions will be found. Therefore, the last two black pixels will only be kept. Similarly, the first black pixel in the case of white–black regions will be kept. The proposed mask has the size of 2 × 4 to fulfill the following two criteria.

1) In this type of a mask, it is divided into three sub masks: The first sub mask is the left mask “2 × 2,” the second sub mask is the center “2 × 1,” and the third sub mask is the right mask “2 × 1,” as marked in Fig. 6. Simply, after each two pixels are checked at once, the first sub mask is applied so that a 2 pixel width “because two column are processed” can be considered for detecting. This process is specified to detect the vertical edges at the intersections of black–white regions. Similarly, the third sub mask is applied on the intersections of white–black regions. Thus, the detected vertical edge has the property of a 1 pixel width.

2) The number “2” points out the number of rows that are checked at once. The consumed time in this case can be less twice in case each row is individually checked.

To select the column at locations (0, 1) and (1, 1) to be the center of the proposed mask, two pixels and one pixel in the case of black–white and white–black regions are retained, respectively.

This process is performed for both of the edges at the left and right sides of the object-of-interest. The first edge can have a black-pixel width of 2, and the second edge can have a black-pixel width of 1. The 2 × 4maskstartsmovingfromtopto bottom and from left to right. If the four pixels at locations (0, 1), (0, 2), (1, 1), and (1, 2) are black, then the other mask values are tested if whether they are black or not. If the whole values are black, then the two locations at (0, 1) and (1, 1) will be converted to white. Otherwise, if column 1 and any other column have different values, the pixel value of column 1 will then be taken. This process is repeated with the whole pixels in the image.

1) Sobel’s Output Form Versus VEDA Output Form: In, the output forms of Sobel and the VEDA are compared. For Sobel, both vertical edges of a detected object have the same thickness, as shown. For the VEDA, there are two- and one-pixel thicknesses for each detected object.

As VEDA’s output form, the searching process for the LP details could be faster and easier because this process searches for the availability of a 2 pixel width followed b ya1pixel width to stand for a vertical edge. This mechanism of searching could save more processing time. In addition, there is no need to search again once the 1 pixel width is faced. These two features could make the searching process faster and easier. A detailed explanation will be provided in the next process [highlighted desired details (HDDs)].

2) Sobel’s Time Complexity versus VEDA’s Time Complexity: By using the big-O-notation module, the code complexity of Sobel and that of the VEDA in terms of time are compared for evaluation.

For the Sobel operator, the algorithm complexity is

$$Os(N) = O(N) \times O(M) \times O(K1) \times O(K2)$$

For the VEDA, the algorithm complexity is

$$Ov(N) = O(N) \times O(M)$$

Where Os(N) and Ov(N) represent the code complexity of the Sobel algorithm and the VEDA, respectively, and O (N) and O(M) represent the code's complexity of the first and second loops inside the Sobel and VEDA codes, respectively. For the Sobel code, O (K 1) × O (K 2) represents the code complexity of the third and fourth loops as deduced from the Sobel code.
In, K 1 and K 2 represent the horizontal and vertical mask scales, respectively. Since these two masks are equal (i.e., 3 × 3), therefore

$$K1 = K2 = K.$$

Thus, the new formulation can be modified as

$$Os(N) = O(N) \times O(M) \times O(K^2).$$

*3) Sobel versus Canny versus VEDA***:** The Sobel operator was the most popular edge detection operator until the development of edge detection techniques with a theoretical basis. It proved popular because it overall gave a better performance than the other contemporaneous edge detection operators. The Canny edge detector has more steps than Sobel. As is known, Canny performs additional processing, i.e., the non maximum suppression that eliminates possible wide ridges that can result from the Canny enhancer or Sobel. Furthermore, where Sobel does simple thresholding, Canny combines the thresholding with contour following to reduce the probability of false contours. Thus, Canny's computation time can be more than Sobel's computation time. We can conclude that Canny is per-haps the most popular edge detection technique at present in terms of the accuracy but costs time. On the contrary, Sobel's accuracy is still accepted in vertical-edge-based applications.
Therefore, the complexity of the canny operator can be found by using big-O-notation, as follows:

$$Oc(N) = O(N) \times O(M) \times O(C^2) + 1.$$

E. HDDs Based on VEDA

After applying the VEDA, the next step is to highlight the desired details such as plate details and vertical edges in the image. The HDD performs NAND– AND operation for each two corresponding pixel values taken from both ULEA and VEDA output images. The NAND– AND procedure for this process is shown. This process depends on the VEDA output in highlighting the plate region. All the pixels in the vertical edge image will be scanned. When there are two neighbor black pixels and followed by one black pixel, as in VEDA output form, the two edges will be checked to highlight the desired details by drawing black horizontal lines connecting each two vertical edges. First, these two vertical edges should be surrounded by a black background, as in the ULEA image. Second, the value of horizontal distance hd represents the length between the two vertical edges of a single object.
The hd has been computed using the test images. The hd value is selected to be suitable for removing long foreground and random noise edges that have not been eliminated earlier. This scanning process will start moving from left to right and from top to bottom. After all pixels are scanned, the regions in which the correct LP exists are highlighted, as shown.

F. CRE

This process is divided into four steps as follows.
1) Count the Drawn Lines per Each Row: The number of lines that have been drawn per each row will be counted and stored in matrix variable HwMny Lines [a] , where a =0, 1 ,...,height-1.
2) Divide the Image into Multi groups: The huge number of rows will delay the processing time in the next steps. Thus, to reduce the consumed time, gathering many rows as a group is used here. Therefore, dividing the image into multi groups could be done using the following:

$$\text{how\_mny\_groups} = \frac{\text{height}}{C}$$

where how_mny_groups represents the total number of groups, height represents the total number of image rows, and C represents the candidate region extraction (CRE) constant. In this paper, C is chosen to represent one group (set of rows). For our methodology, C = 10 because each ten rows could save the computation time. In addition, it could avoid either loosing much desired details or consuming more computation time to process the image.
Therefore, each group consists of ten rows. Due to the HwMnyLines[a] values, some rows have a number of drawn horizontal lines, and this makes some groups to have horizontal lines. The step here is to store the total number of horizontal lines for each group. In this step, a matrix is created to store the total number of drawn lines for each ten rows (group). This step will distinguish the regions that might have LP details.
In, [group] represents values from 0 to 28. This range is calculated using as

$$\text{how\_mny\_groups} = \frac{288}{10} \approx 29$$

where the total number of image rows is equal to 288, and the total number of groups is equal to 29 groups.
This figure denotes that the highest values of line frequency are located in the LP region.
*3) Count and Store Satisfied Group Indexes and Boundaries***:**

Most of the group lines are not parts of the plate details. Therefore, it is useful to use a threshold to eliminate those unsatisfied groups and to keep the satisfied groups in which the LP details exist in. Each group will be checked; if it has at least 15 lines, then it is considered as a part of the LP region. Thus, the total number of groups including the parts of LP regions will be counted and stored. The remaining groups after thresholding step should have the LP details. Therefore, their locations are stored. The final step here is to extract both upper and lower boundaries of each satisfied group by using its own index. This threshold (i.e., 15 lines) value is determined to make sure that the small-sized LP is included for a plate searching process. If the predefined threshold is selected with a less value than that given, wrong result can be yielded because noise and/or non plate regions will be considered as parts of the true LP, even if they are not. Based on that, the optimal threshold for the best detection rate has been found is $\geq 1/20 \times$ image_height. Therefore, the threshold in our case is $\geq 1/20 \times 288 \geq 14.4 \equiv 15$.

4) Select Boundaries of Candidate Regions: This step draws the horizontal boundaries above and below each candidate region shows the result of drawing candidate regions boundaries in the input image. As shown, there are two candidate regions interpreted from horizontal-line plotting, and these conditions require an additional step before the LP region can be correctly extracted.

G. PRS

This process aims to select and extract one correct LP.The process is discussed in five parts. The first part explains the selection process of the LP region from the mathematical perspective only. The second part applies the proposed equation on the image. The third part gives the proof of the proposed equation using statistical calculations and graphs. The fourth part explains the voting step. The final part introduces the procedure of detecting the LP using the proposed equation. The flowchart of plate region selection (PRS) and plate detection (PD) is also provided, as shown.

1) Selection Process of the LP Region: As known, some of the processed images are blurry, or the plate region might be defected. The plate region can be checked pixel by pixel, whether it belongs to the LP region or not. A mathematical formulation is proposed for this purpose, and once this formulation is applied on each pixel, the probability of the pixel being an element of the LP can be decided.

As aforementioned, for the candidate regions, each column will be checked one by one. If the column blackness ratio exceeds 50%, then the current column belongs to the LP region; thus, this column will be replaced by a vertical black line in the result image, as shown. Hence, each column is checked by the condition that, if blck Pix$\geq 0.5\times$ colmnHght, then the current column is an element of the LP region. Here, the blck Pix represents the total number of black pixels per each column in the current candidate region, and the colmnHght represents the column height of the of the candidate region. This condition with a fixed value (0.5) is used with non blurry images. However, some pixels of the candidate regions will not be detected in case the ratio of blackness to the total length (height) of the candidate region is greater than 50%. Therefore, the condition is changed to be less than 50%, according to the ratio of the blurry level or the deformation of the LP. The condition will be modified as follows. blckPix $\geq P_{RS} \times$ colmnHght, where P_{RS} represents the P_{RS} factor. The P_{RS} value is reduced when the blurry level is high to highlight more important details, and it is increased when the blurry level is less. Therefore, the mathematical representation for selecting the LP region can be formulated as follows:

$$C_{\text{region}} = \begin{cases} 0, & \text{blckPix} \geq P_{\text{RS}} \times \text{colmnHght} \\ 255, & \text{otherwise} \end{cases}$$

where C_{region} represents the output value for the current pixel of the currently processed candidate region. If $C_{region} = 0$, consider the checked pixel as an element of the LP region; otherwise, consider it as background. The value of P_{RS} is automatically determined. Below is the proposed pseudo code is applied.

If (blckPix> = 0.5× colmhHght)
//i.e., PRS = 50 %; the blckPix> = 0.5× the column height of the correct region
Then
PRS is determined, and = 0.5 →the current column is apart of the LP;
Else If (blckPix> = 0.4× colmhHght)
//i.e., PRS = 40 %; the blckPix> = 0.4× the column height of the correct region
Then
PRSis determined, and = 0.4 →the current column is apart of the LP;
Else If (blckPix> = 0.3× colmhHght)
//i.e., PRS = 30 %; the blckPix> = 0.3× the column height of the correct region
Then
PRSis determined, and = 0.3 →the current column is apart of the LP;
Else
{
The current column is not a part of the LP, and it belongs to background; Go to next column;
}
End If

2) Applying the Mathematical Formulation: After applying on the image that contains the candidate regions shown.

3) Proof of (12) Using Statistics: This shows the blackness frequency for both candidate region s shown. From the plot, it can be observed that the true candidate region has higher blackness frequency than the wrong candidate.

4) Making a Vote: The columns whose top and bottom neighbors have high ratios of blackness details are given one vote. This process is done for all candidate regions. Hence, the candidate region that has the highest vote values will be the selected region as the true LP. By tracking the black vertical lines, the plate area will be detected and extracted. In, the boundaries of the plate area are detected.

After the whole procedure has been performed, the final result for LPD.

5) LPD: An example is given to show how the value of the P RS factor can whether increase or decrease the detection rate of the CLPD method. After applying on any image obtained from the previous process (i.e., CRE), the output image contains a single candidate region. This effect of changing the PRS value is given in the example. In, the boundaries of the plate area are detected, and the corresponding LP is marked, as shown whereas the P_{RS} value in this case has been set to be greater than or equal to 0.5. In the P_{RS} value is set to be greater than or equal to 0.4. In the last case, the P_{RS} value is set to be greater than or equal to 0.3.By comparing the three obtained results, we can say that the best result is shown. Thus, by controlling the P_{RS} value based on the processed image, the performance of the CLPD method is enhanced.

V. EXPERIMENTAL RESULTS AND DISCUSSION

Here, the AT process will be evaluated first. Then, the accuracy and the computation time of the VEDA are compared with that of the Sobel operator. Finally, the performance of the proposed CLPD method is evaluated. To carry out this evaluation and analysis, The VEDA and the Sobel algorithm are separately used to extract vertical edges. In the first evaluation, the CLPD method has been built as follows: ULEA→VEDA →HDD→ CRE →PRS →PD. In the second evaluation, the CLPD method has been built as follows: ULEA →Sobel →HDD→CRE → PRS →PD. Our proposed method has been tested on the laptop with the following specifications: Core 2 CPUs with 1.83 GHz and 1 GB of RAM. The program is running under Windows XP SP2 written in C++ language with an Open CV library.

A. AT Evaluation

To evaluate the effect of T mentioned on the performance of the CLPD method, the AT has been applied on the image shown, with different values of T.

As shown, when T is equal to 0.04, there are adjacent regions around the LP. In some important details such as the LP characters are partially eliminated and changed to background when the T is 0.3. Thus, the detection rate of the CLPD method might be affected in such a case.

B. VEDA versus Sobel

1) Accuracy: Both Sobel and VEDA have been running on same software and hardware environments to compare the accuracy, computation time, and complexity.

The accuracy and the computation time of the VEDA and Sobel are shown. As stated, the VEDA consumed only 11% of Sobel's time, yet the appearance of edges is comparable.

*2) Computation Time :*This shows the computation time of the VEDA and Sobel for different image sizes. The best performance of VEDA is one ninth the computational times compared with Sobel. The ratio is getting less when the resolution of image increases.

3)Big-O-Notation-Module-Based Complexity Comparison:

As discussed earlier in the proposed CLPD method, the VEDA has less complexity than the Sobel operator by K^2times.Therefore, the VEDA is easier and a very simple tool to be used for extracting vertical edges.

C. Proposed CLPD Method Evaluation

Here, the samples collection will be explained. Then, the CLPD method will be evaluated using the VEDA and Sobel for extracting vertical edges. The proposed CLPD method will be compared with the Malaysian CAR Plate Extraction Technology (CARPET) and several methods in terms of the computation time and the detection rate.

1) Sample Collection: The whole samples used in our experiments were captured in different parking areas and under various weather conditions such as rainy, sunny, and shady days taken from 8:00 A . M .to6:00 P. M. They contain different backgrounds and objects such as trees and two LPs. A lot of difficulties in the experiments are faced such as blurriness, illumination changes, similarity between LP and car-body colors, LP sizes, LP designs, and double-row LPs, as shown gives the experimental conditions. If the proposed algorithm is applied on a higher specification camera, the camera distance will also be increased.

2) *CLPD Method Evaluation Using the VEDA*: The whole tested images are 664, and they are classified into three groups based on the images quality, background complexity, and lighting effects. The first group (G1) includes the ideal, blurry, and defected plates. The second group (G2) contains double-row LPs, trees, similarity of LP and car body, and more than one LP or a car in a single image. The third group (G3) contains excessive light, rainy, shady, and low-contrast images shows the number of images per each classified group. To show the increment percentage of the detection rate shows the detection rate of each group before and after applying. As shown from the results of this figure, the increment percentage of the detection rate after applying has been raised from 86.3% to 91.65% for all test images shows the total number of success and failure detection for each group after applying. The detection rate for G2and G3 is lower than G1 due to the complexity existing in the image shows the percentage of success and failure for each group shows the average processing times for all test images. Most of the processed images take 47 ms. The computation time for each of the seven stages in the proposed method is listed. A lot of the time

is consumed on the second stage, which is AT. The total time of processing one 352 × 288 image is 47.7 ms, which meets the real-time processing requirement. The stage that requires the longest time to compute is AT. The Combination of the ULEA and the VEDA still consumed less time than AT.

As summarized, 607 LPs from the 664 images are successfully detected. Then, the average accuracy of CLPD is 91.4%. In addition to the successfully detected LPs shown, a set of further samples that are taken randomly is shown, and their located LPs are marked. It can be seen that the proposed method is able to detect LPs, although the LP is tilted. In addition, if there are many cars or LPs on a single image, the proposed CLPD method can detect them and extract the correct LPs, as shown.

A comparison of the proposed CLPD method with several methods will be provided. Additionally, the computation time of the proposed CLPD method is compared with the computation time of other methods.

As shown, our proposed CLPD method can be used with real-time applications, and it has the lowest computation time compared with several methods.

3) CLPD Method Evaluation Using Sobel: In this case, the following steps have been used: ULEA →Sobel →HDD→ CRE →PRS →PD summarizes the computation time and the detection rate of the proposed CLPD method.

Based on these results, it is concluded that using the VEDA for detecting vertical edges could have enhanced the performance of the proposed CLPD in terms of the computation time and the detection rate.

Fig.2 Input Image

Fig.3 AT image

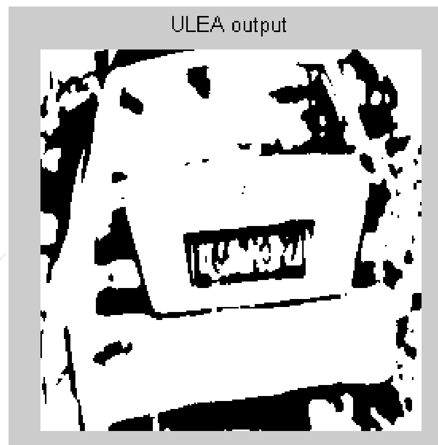

Fig.4 ULEA output

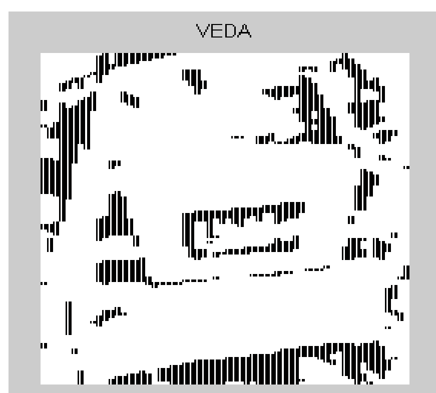

Fig.5 VEDA output

TABLE 1. COMPUTATION TIME AND DETECTION RATE OF THE CLPD METHOD USING VEDA AND SOBEL

Evaluation	No of Correctly detected plates	Detection Rate	Computation Time (ms)
VEDA based	607/664	91.4%	41.7
SOBEL based	591/664	89%	101.7

VI. CONCLUSION AND FUTURE WORK

The new and fast algorithm for vertical edge detection, in which its performance is faster than the performance of Sobel by five to nine times depending on image resolution. The VEDA contributes to make the whole proposed CLPD method faster. The CLPD method in which data set was captured by using a web camera. This system employed 664 images taken from various scenes and under different conditions. Only one LP is considered in each sample for the whole experiments. In the experiment, the rate of correctly detected LPs is 91.4%. In addition, the computation time of the CLPD method is 47.7 ms, which meets the real-time requirements. Finally, the VEDA-based and Sobel-based CLPD are compared, and the findings show that VEDA-based CLPD is better in terms of the computation time and the detection rate.

REFERENCES

[1] Al-Ghaili.A.L, Mashohor.S, Ismail.S, and Ramli.A .R, “A new vertical edge detection algorithm and its application,” in Proc. IEEE Int. Conf.Comput. Eng. Syst., Cairo, Egypt, 2008, pp. 204–209.

[2] Bai.H and Liu.C, “A hybrid license plate extraction method based on edge statistics and morphology,” in Proc. 17th Int. Conf. Pattern Recognit., Cambridge, U.K., 2004, pp. 831–834.

[3] Fukumi.M, Takeuchi.Y, Fukumoto.H, Mitsura.Y, and Khalid.M,“Neural network based threshold determination for Malaysia license plate character recognition,” in Proc. 9th Int. Conf. Mechatron. Technol., 2005,pp. 1–5.

[4] Huda S.N, Marzuki.K, Rubiyah.Y, and Khairuddin.O, “Comparison of feature extractors in license plate recognition,” in Proc. 1st IEEE AMS, Phuket, Thailand, 2007, pp. 502–506.

[5] Lee.E.R, Pyeoung.K.K, and Hang.J.K, “Automatic recognition of a car license plate using color image processing,” in Proc. IEEE Int. Conf. Image Process., 1994, pp. 301–305.

[6] Naito.T,Tsukada.T, Yamada.K, Kozuka.K, and Yamamoto.S, “Robust license-plate recognition method for passing vehicles under outside environment,” IEEE Trans. Veh. Technol., vol. 49, no. 6, pp. 2309–2319, Nov. 2000

[7] Paris.Ri, Di.E.D Claudio, Lucarelli.G, and Orlandi.G, “Car plate recognition by neural networks and image processing,” in Proc. IEEE Int. Symp. Circuits Syst., 1998, pp. 195–198.

[8] Thanongsak.S and Kosin.C, “The recognition of car license plate for automatic parking system,” in Proc. 5th Int. Symp. Signal Process. Appl., Brisbane, QLD, Australia, 1999, pp. 455–457.

[9] Wu.H.H.P, Chen.H.H, Wu.R.J, and Shen.D.F, “License plate extraction in low resolution video,” in Proc. IEEE 18th Int. Conf. Pattern Recognit., Hong Kong, 2006, pp. 824–827.