International Journal of Scientific Research in Computer Science and Engineering
Vol.5, Issue.6, pp.19-26, December (2017)

Research Paper

E-ISSN: 2320-7639

Car License Plate Detection Using Veda

M. Bhargavi[1*] and Sajja.Radharani[2]

[1*]Dept of CSE, VFSTR, Guntur, India
[2]Dept of CSE,VFSTR, Guntur, India

[*]*Corresponding Author: bhargaviformal@gmail.com*

Available online at: www.isroset.org

Received: 07/Nov/2017, Revised: 19/Nov/2017, Accepted: 14/Dec/2017, Published: 31/Dec/2017

***Abstract*—** The VEDA is one of the emerging techniques in the field of Image Processing.Car-license-plate detection (CLPD) comprises of two main steps. The first step comprises of converting a color image into grayscale values, which raises the speed of the CLPD method. Then input to Adaptive threshold (AT) to segment the image and binarised it, then apply unwanted-line elimination algorithm (ULEA) to remove unwanted lines thus enhance the image, then VEDA is applied. The second part comprises of CLPD method which processes low resolution images taken by a web camera. Now the VEDA detects the vertical edges,the desired plate details based on color information are highlighted. Now,performing statistical and logical operations the candidate region will be extracted. Finally,the license plate (LP) is detected.

***Keywords*—** Car-License Plate Detection(CLPD),Vertical Edge Detection Algorithm(VEDA), Unwanted-Line Elimination Algorithm(ULEA), AdaptiveThresholding(AT), LicensePlate(LP).

I. INTRODUCTION

The car-license-plate recognition system is an image processing technology used to identify vehicles by capturing their car license plates. The car-license-plate recognition technology is known as automatic number plate recognition, automatic vehicle identification, car-license-plate recognition, or optical character recognition for cars. The car-license-plate detection and recognition system became an important area of research due to its various applications

License plate numbers are being used to uniquely identify a vehicle. License plate recognition system plays an important role in many applications like electronic payment system and parking fee, to find stolen cars , traffic surveillance. For example in parking, number plates are used to calculate the duration of the parking. When a vehicle enters the gate, license plate is automatically recognized and stored in database. On leaving, the license plate is recognized again and compared with the stored numbers in the database. The time difference is used for calculating the parking fee[1],[2]. License plate recognition is convenient and cost efficient as it is automated

There are several License plate detection methods that have been used before, such as morphological operations, the edge extraction combination of gradient features ,salient features a neural network for color or grayscale classification, and vector quantization.Due to ambient lighting conditions, interference characters, and other problems, it is difficult to detect LPs in complex conditions. Some of previous license plate detection methods are restricted to work under certain conditions, such as fixed backgrounds and known color.

The Organization of the paper is as follows,Section I contains the introduction of car license plate detection and feature extraction,Section II contains the literature survey ,Section III contains the existing system that is LP detection using morphological and neural networks,Section IV contains the proposed method to detect LP efficiently,Section V describes the results and Section VI contains conclusion.

II. LITERATURE SURVEY

This chapter deals with survey of various research papers that have contributed in the Detection of car license

A)A hybrid License Plate Extraction Method Based on Edge Statistics and Morphology.

In the system, the different threshold of the license plate location is regarded as the different scale. The thresholds are 64, 32, 16, 8, which are the first, the second, the third, the fourth scale. Bigger scale, the number of the FPs is small, the run time of system is little, but the license plate is detected hard. In the low scale, the more FPs, long time, more. It is based on the edge statistic and morphology[3]. The proposed approach can be divided into four sections, which are, the vertical edge detection, the edge statistical analysis, the hierarchical-based license plate location, and the morphology-based license plate extraction. The algorithm gives good results on our database, and it is relatively robust to variations of the lighting conditions and different kinds of

vehicle. From the result of experiment, the scheme is satisfying.

After the preprocessing, following steps gets the candidate regions of morphology-based license plate location:

1) Connected Component Analysis

The connected component analysis algorithm is applied to the processed images. So we get the bounding rectangle of the object and the number of the object pixels in these rectangles.

2) Feature Extraction

With the important information from the CCA, some features of region, such as the aspect ratio (R), the area (A) and the density (D) of region are applied. Let Re denote the region of rectangles with width W and height H, then R =W / H and A =W × H . Let N denote of the number of the object pixels in the rectangles, then D = N /(W × H) . The result by using these features is that most of components are deleted, and 1-5 candidate region(s) will be gotten.

3) Combination of candidate regions

The step can refer to the section B.

4) Getting Final Candidate regions

Some more strict conditions will be applied, and then we can get the final position of license plate, the number of candidates is less than 3. In general, the number is only 1.

Merits:

It is the robust and simple method.

Demerits:

The accuracy of the detection is low.

B)Neural Network Based Threshold Determination for Malaysia License Plate Character Recognition

In this paper, we present a vehicle license plate character recognition system and its simulation results. In character recognition, a threshold value for separating characters and background is in particular important. We determine a threshold value by using a three-layered neural network (NN).

Furthermore, in extracting character portions, characters are segmented by character information, and then recognized by obtaining their directional features and by using NN. The next section describes the neural network for determining a threshold value of binarization , and explains the technique of image processing for performing character recognition. This is followed by discussions on the experimental results and conclusion[4]. In addition, inter-frame difference can perform the car extraction from a video easily.

Since plate location can be mostly decided on the basis of the vehicle type after car location by the inter-frame difference, preliminary experiments using images with roughly detected plate regions are conducted. For better segmentation of license plate characters, we have to evaluate a method by two thresholds (upper and lower bounds) determined by neural networks or genetic algorithms . Furthermore, we will evaluate a statistical feature generation method, Fisher linear discriminant analysis (FLDA). The FLDA is basically better than the principal component analysis in feature generation.

Merits:

The Neural network classifier will recognize the plate better than the Existing Classifier.

Demerits:

The feature extraction of this method have to improved.

C)License Plate Extraction in Low Resolution Video

The license plate is detected in the low resolution video. Qualities of the Images taken outdoors are usually influenced by the weather and the environment. For example, the position of the sun can generate images with widely variant brightness. Moreover, the roadside buildings and trees might cast the shadow on the vehicle and further reduce the contrast of the input image[5]. Hence, contrast enhancement is needed to increase the image quality and improve the result of LP detection. 'Bottom-hat' operation can enhance various kinds of texture, many non-LP regions also show up significantly; for example, the frames of the vehicle and its windows. However,the region of the LP usually contains denser structure than those of others due to the close arrangement of the alphanumeric data. Fortunately, morphological gradient can be used to distinguish a denser region from a looser one. The closed image is then binarized by Otsu's algorithm and all connected components are labeled as LP candidates.

After the detection process, the system generated binary image with labeled LP candidates. Each one of the candidates must be checked by certain criteria to see if it satisfies the requirements of a LP. Two criteria are adopted sequentially, the geometric properties of the LP is utilized first, and then characteristics of the alpha-numerals in the LP are employed in the process of verification.

Merits:

The method detects the license plate in the low resolution video. The Existing does not detect the license plate in the low resolution images.

Demerits:

The method takes more time for the execution.

D)Morphology-based License Plate Detection from Complex Scenes

The proposed license plate detection technique can locate multiple plates with different orientations. The proposed system is composed of three major parts: feature extraction, detection of license plate candidates, and license plate verification. The license plate is a pattern composed of several characters, which have high contrast changes to their background. In this paper, we use several morphological operations to find the high contrast area as important features to detect license plates [6]. Before introducing the proposed method, some morphological operations should be introduced first. After labeling, a set of potential license plates can be

extracted from the images. However many incorrect license plates may be extracted from the cluttered environment. For instance, frames of windows, trees, edges among a set of books, etc. are frequently segmented as license-plate-analogue pixels. Therefore, some geometries and texture information are used first at this stage to remove these unwanted regions. Three criteria are defined here for eliminating impossible license plates. Let R denote the extracted plate region with the size w× h . The first criterion is the density of the region R: den = A/(h×w) , where A is the area of R. The second criterion is the ratio r between the width and height of R.. The third criterion is that the size of a license plate should be larger than a fixed size, for example, 60× 25 . If the size of a license plate is not large enough, the characters in the license plate will be too small for recognition. These criteria will significantly reduce the number of potential license- plate-analogue segments into few candidates.

Merits:
It is a simple and easy implementation method.

Demerits:
The methods not perform well for the low resolution images.

E)An Approach to Korean License Plate Recognition Based on Vertical Edge Matching
Here we introduce the method to recognize the License plate for the Korean images. The proposed algorithm is fast enough [7]. The recognition unit of an LPR system can be implemented only in software so that the cost of the system is reduced. For the binary image {E,,,}, the size-and-shape filter based on seed filling algorithm is described as follows:
1) Search the entire image row by row, for each white pixel in the image, if it has not been checked, then run over the eight connected white regions by using seed filling algorithm in which it is adopted as the first starting seed in the region.
2) Check whether the region have specified features. If it does not satisfy some predefined restricted conditions, then fill the region with black, that is, remove the region as noise, since it is impossible to be the region of interest (ROI); Otherwise left it in the image as a possible interested object and record its information for post-processing. After that, mark whole pixels of this region as check pixels.
3) Continue to scan the image row by row to find another unchecked white pixel as the first starting seed of a new region, until all white pixels in the image have been checked.
In Korean license plate extraction, after the vertical edge image has been obtained, it is filtered so as to remove the edges that are impossible to be the vertical edge of a license plate. Before filtering, morphological operation “dilation” is applied as a pre-processing. In the vertical edge image, an edge area is defined as a se1 of white pixels that are eight connected neighbors with each other. For each edge area, we check its size and shape by the seed filling based filter.

If the edge area is smaller than a predefined size, or does not form a beeline whose slope is within a predefined interval, then it is removed, since it is impossible to be the vertical edge of a license plate. Otherwise, the coordinates of the top and bottom pixels of the edge area are recorded for edge matching in post-processing. Since the vertical edges of a license plate may be cut off in the vertical edge image.

Merits:
The detection rate is high than the existing system.

Demerits:
The method does not work for the other country license plate.

III. EXISTING SYSTEM

In Existing system different kinds of the methods are used such as morphological operation and Neural network.

A) Morphological Operation:
Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. According to Wikipedia, morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to grayscale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest.

Morphological techniques probe an image with a small shape or template called a structuring element[8]. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels. Some operations test whether the element "fits" within the neighborhood, while others test whether it "hits" or intersects the neighborhood:

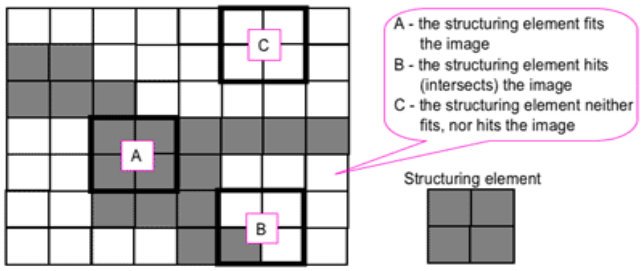


Fig 3.1:Probing of an image with a structuring element (white and grey pixels have zero and non-zero values, respectively).

A morphological operation on a binary image creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image.The structuring element is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one:

- The matrix dimensions specify the size of the structuring element.
- The pattern of ones and zeros specifies the shape of the structuring element.
- An origin of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.

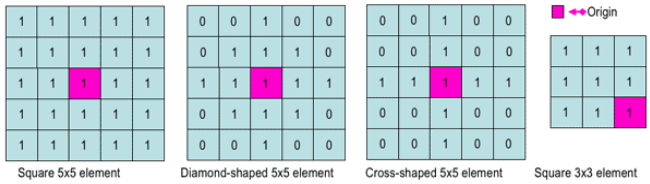
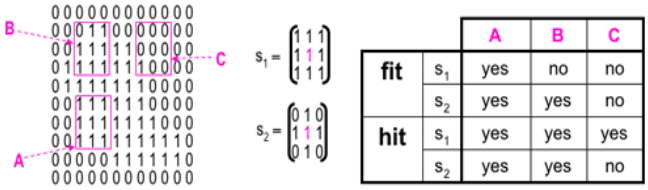


Fig 3.2 :Examples of simple structuring elements.

A common practice is to have odd dimensions of the structuring matrix and the origin defined as the center of the matrix. Structuring elements play in morphological image processing the same role as convolution kernels in linear image filtering.

When a structuring element is placed in a binary image, each of its pixels is associated with the corresponding pixel of the neighborhood under the structuring element. The structuring element is said to fit the image if, for each of its pixels set to 1, the corresponding image pixel is also 1. Similarly, a structuring element is said to hit, or intersect, an image if, at least for one of its pixels set to 1 the corresponding image pixel is also 1.



		A	B	C
fit	s_1	yes	no	no
	s_2	yes	yes	no
hit	s_1	yes	yes	yes
	s_2	yes	yes	no

Fig 3.3:Fitting and hitting of a binary image with structuring elements s_1 and s_2.

Zero-valued pixels of the structuring element are ignored, i.e. indicate points where the corresponding image value is irrelevant.

B) Neural Network

Neural Network ?

The area of Neural Networks probably belongs to the borderline between the Artificial Intelligence and Approximation Algorithms. Think of it as of algorithms for "smart approximation". The NNs are used in (to name a few) universal approximation (mapping input to the output), tools capable of learning from their environment, tools for finding non-evident dependencies between data and so on.

The Neural Networking algorithms (at least some of them) are modeled after the brain (not necessarily - human brain) and how it processes the information. The brain is a very efficient tool. Having about 100,000 times slower response time than computer chips, it (so far) beats the computer in complex tasks, such as image and sound recognition, motion control and so on. It is also about 10,000,000,000 times more efficient than the computer chip in terms of energy consumption per operation.

The brain is a multilayer structure (think 6-7 layers of neurons, if we are talking about human cortex) with 10^11 neurons, structure, that works as a parallel computer capable of learning from the "feedback" it receives from the world and changing its design (think of the computer hardware changing while performing the task) by growing new neural links between neurons or altering the activities of existing ones. To make the picture a bit more complete, let's also mention, that a typical neuron is connected to 50-100 of the other neurons, sometimes, to itself, too.

To put it simple, the brain is composed of neurons, interconnected.

Structure of a neuron.

Our "artificial" neuron will have inputs (all N of them) and one output:

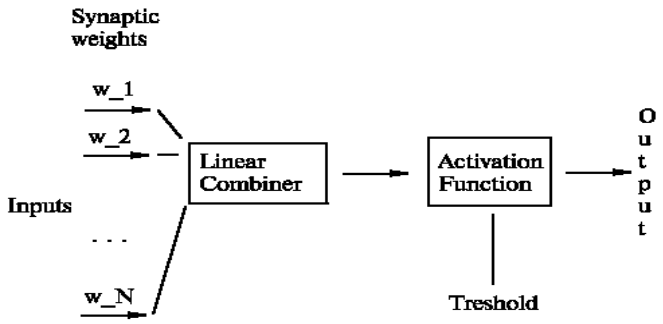


Fig 3.4.Structure of a Neuron

As you can see the neuron has set of nodes that connects it to Inputs, output or other neurons these nodes are also called synapses. A linear combiner, which is a function that takes all inputs and produces a single value. A simple way of doing it is by adding together the dInput (in the case if you are not a programmer - a "d" prefix means "double", we use it so that the name (dInput) represents the floating point number) multiplied by the Synaptic Weight dWeight:

```
for(int i = 0; i < nNumOfInputs; i++)
 dSum = dSum + dInput[i] * dWeight[i];
```

An Activation Function. We do not know what the Input will be. Consider this example - the human ear can function near the working jet engine and at the same time - if it was only

ten times more sensitive, we would be able to hear a single molecule hitting the membrane in our ears! What does that mean? It means that the input should not be linear. When we go from 0.01 to 0,02, the difference should be comparable with going from 100 to 200.

How do we make a non-linear input? By applying the Activation function. It will take ANY input from minus infinity to plus infinity and squeeze it into the -1 to 1 or into 0 to 1 interval.

Finally, we have a threshold. What the INTERNAL ACTIVITY of a neuron should be when there is no input? Should there be some threshold input before we have the activity? Or should the activity be present at some level (in this case it is called a bias rather than a threshold) when the input is zero?

For simplicity, we (as well as the rest of the world) will replace the threshold with an EXTRA input, with a weight that can change during the learning process and the input is fixed and always equal (-1). The effect, in terms of mathematical equations, is exactly the same, but the programmer has a little more breathing room.

IV. PROPOSED SYSTEM

It is the novel method for detecting the license plate in the image. In the initial step we have to denoise the image. And then we segment the image by the use adaptive threshold segmentation. Then we eliminate the unwanted lines. Adaptive threshold is the simplest non-contextual segmentation technique. With a single threshold, it transforms a grayscale or color image into a binary image considered as a binary region map. The binary map contains two, possibly disjoint regions, one of them containing pixels with input data values smaller than a threshold and another relating to the input values that are at or above the threshold. The previous methods use more sophisticated web cameras so the process is very difficult and time consuming. We use a web camera of 352 × 288 resolutions. The web cameras are processed offline. Vertical edge extraction and detection is an important process in CLPD because it affects the system's accuracy and computation time. Our proposed vertical edge detection algorithm reduces these complexities. The color images are converted into gray scale for easy processing

The system Architecture of the Vertical edge based detection algorithm has been divided into five modules,and the architecture has been given below

MODULES:

A. Image preprocessing.
B. Adaptive Thresholding.
C. Unwanted Line Elimination.
D. Vertical edges detection.
E. Plate region Detection

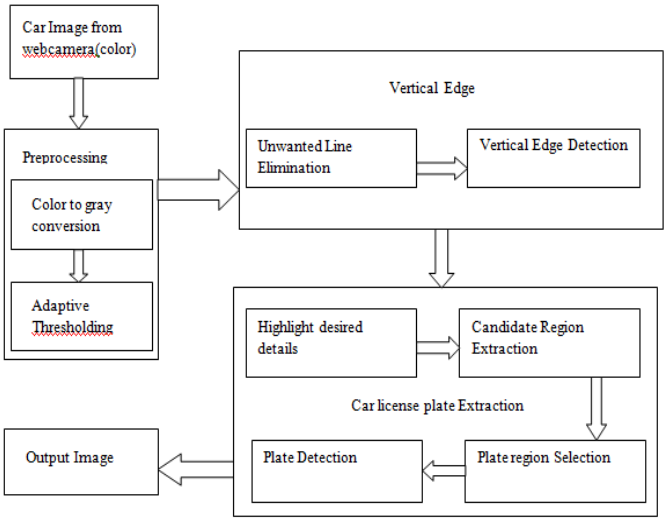


Fig 4.1:System Architecture

A) Preprocessing:

In preprocessing method median filter is used to remove noise from the input test images. It is often desirable to be able to perform some kind of noise reduction on an image or signal. The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing. Median filtering is very widely used in digital image processing. The computational effort and time is spent on calculating the median of each window. Because the filter must process every entry in the signal, for large signals such as images, the efficiency of this median calculation is a critical factor in determining how fast the algorithm can run. Median filtering is one kind of smoothing technique, as is linear Gaussian filtering. All smoothing techniques are effective at removing noise in smooth patches or the smooth regions of a signal, but adversely affect edges. Often, though, at the same time as reducing the noise in a signal, it is important to preserve the edges. Edges are of critical importance to the visual appearance of images, for example. For small to moderate levels of (Gaussian) noise, the median filter is demonstrably better than Gaussian blur at removing noise whilst preserving edges for a given, fixed window size

B) Adaptive Thresholding:

Our adaptive thresholding technique is a simple extension of Wellner's method [Wellner 1993]. The main idea of Wellner's algorithm is that each pixel is compared to an average of the surrounding pixels. Specifically, an approximate moving average of the lasts pixels seen is calculated while traversing the image. If the value of the current pixel is t percent lower than the average then it is set to black, otherwise it is set to white. This method works

because comparing a pixel to the average of nearby pixels will preserve hard contrast lines and ignore soft gradient changes. The advantage of this method is that only a single pass through the image is required. Wellner uses a $1/8^{th}$ of the image width for the value of s and 15 for the value of t. However, a problem with this method is that it is dependent on the scanning order of the pixels.

In addition, the moving average is not a good representation of the surrounding pixels at each step because the neighborhood samples are not evenly distributed in all directions. By using the integral image (and sacrificing one additional iteration through the image), we present a solution that does not suffer from these problems. Our technique is clean, straightforward, easy to code, and produces the same output independently of how the image is processed. Instead of computing a running average of the last s pixels seen, we compute the average of an s x s window of pixels centered around each pixel. This is a better average for comparison since it considers neighboring pixels on all sides. The average computation is accomplished in linear time by using the integral image. We calculate the integral image in the first pass through the input image. In a second pass, we compute the s x s average using the integral image for each pixel in constant time and then perform the comparison. If the value of the current pixel is t percent less than this average then it is set to black, otherwise it is set to white.

C) Unwanted Line Elimination:

Thresholding process in general produces many thin lines that do not belong to the LP region. These lines may interfere with the LP location.

There are four cases in which unwanted lines can be formed.

i) In the first case, the line is horizontal by an angle equal to 0◦ .
ii) In the second case , the line is vertical by an angle equal to 90◦ .
iii) In the third case, the line is inclined by an angle equal to 45◦.
iv) In the fourth case, the line is inclined by an angle equal to 135◦.

The black pixel values are the background, and the white pixel values are the foreground. Only black pixel values in the threshold image are tested.

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.

D) Vertical Edge Detection:

The advantage of the VEDA is that it differentiates the plate detail region. A 2 × 4 mask is proposed for this process, the center pixel of the mask is located at points (0, 1) and (1, 1). The 2 × 4 mask starts moving from top to 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 they are black, then the two locations at (0, 1) and (1, 1) are 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. The edges extracted from a two-dimensional image of a three-dimensional scene can be classified as either viewpoint dependent or viewpoint independent[9].

A viewpoint independent edge typically reflects inherent properties of the three-dimensional objects, such as surface markings and surface shape. A viewpoint dependent edge may change as the viewpoint changes, and typically reflects the geometry of the scene, such as objects occluding one another. A typical edge might for instance be the border between a block of red color and a block of yellow. In contrast a line (as can be extracted by a ridge detector) can be a small number of pixels of a different color on an otherwise unchanging background. For a line, there may therefore usually be one edge on each side of the line.

E) Plate Region Detection:

It contains three steps. Count the Drawn Lines per Each Row

1)Divide the Image into Multi groups : An Image consists of a large number of rows and columns of pixels ,so the image is sub-divided into groups.
2)Count and Store Satisfied Group Indexes and Boundaries : Some of the groups may be a not a part of car number(candidate region) these groups were eliminated.
3)Select Boundaries of Candidate Regions : Horizontal lines are drawn at the borders of the candidate region.
4)Select the LP Region : The plate region can be checked pixel by pixel and by using blackness ratio . If the ratio is more than 50%, then the column belongs to LP.

Apply Mathematical Formulas

a) Prove Using Statistics : candidate region has a higher blackness frequency than the wrong candidate.
b) 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.
c) LPD : Using the PRS factor.

V. PERFORMANCE METRICS

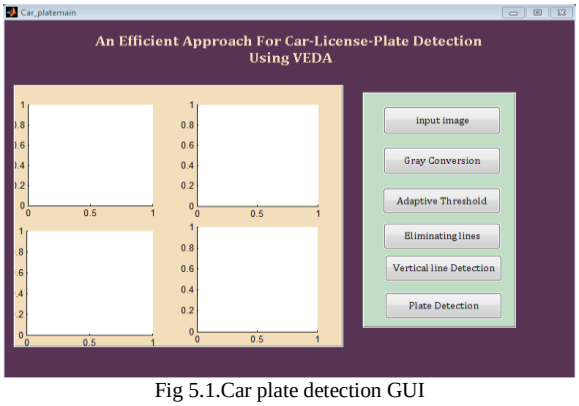


Fig 5.1.Car plate detection GUI

The Fig 5.1 is the initial GUI for the vertical edge based car license plate detection method

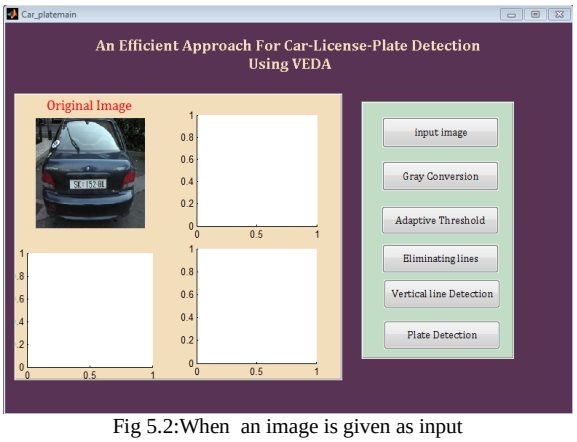


Fig 5.2:When an image is given as input

The fig 5.2,When the input image button is pressed then in the first axis the Original Image is displayed

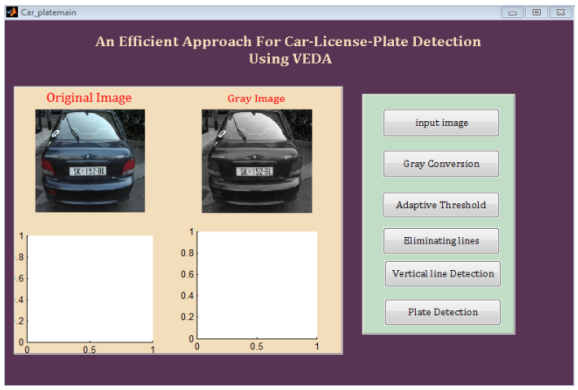


Fig 5.3:input image converted as Gray image

When the first button is pressed the input image is taken later on we have to press the gray conversion which is used to convert the original image into gray image that is shown in fig 5.3

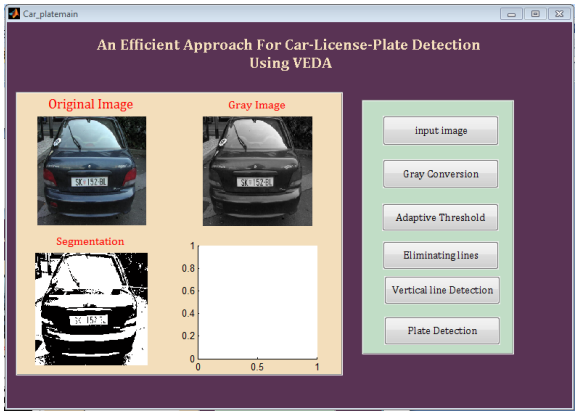


Fig 5.4.Adaptive Threshold applied on gray image

After the gray conversion of the original image we have to apply the adaptive threshold on the converted image ,here the adaptive threshold button is pressed the image is segmented and it is shown in the fig 5.4

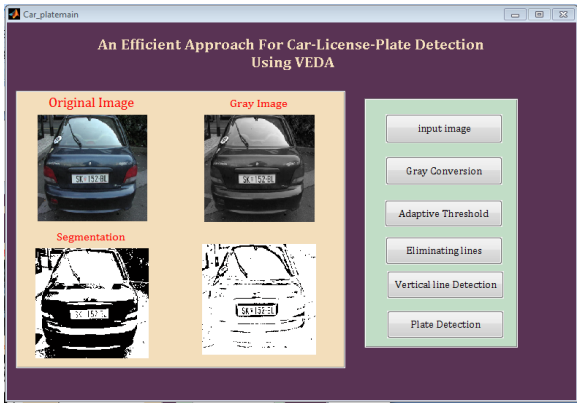
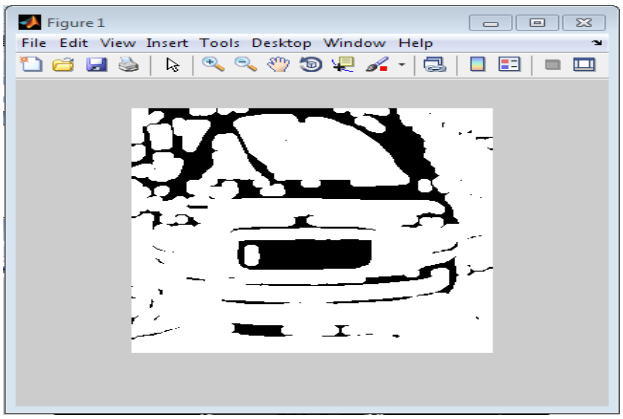
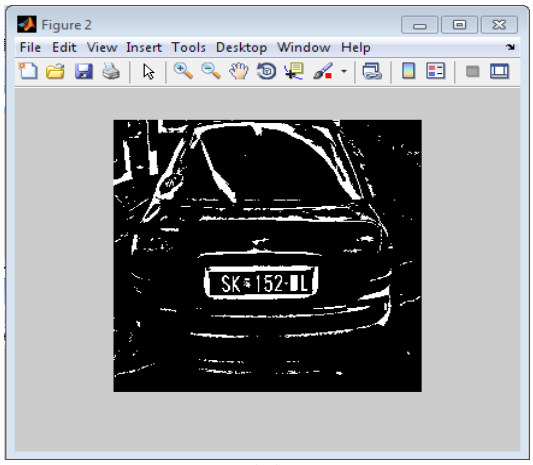


Fig 5.5. Eliminating unwanted lines

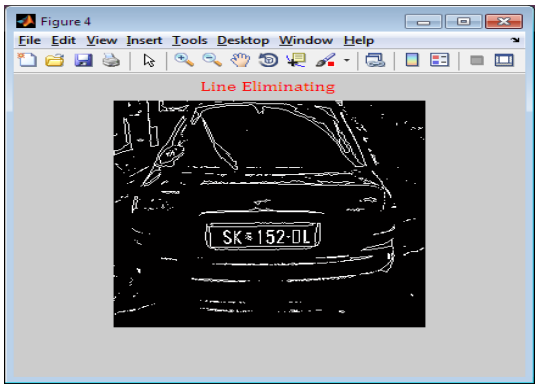
After applying the adaptive threshold we have to eliminate the unwanted line in the segmented image for that we have to press the Eliminating lines button then the unwanted lines are eliminated from the image that are shown in the above and below figures.



(A)



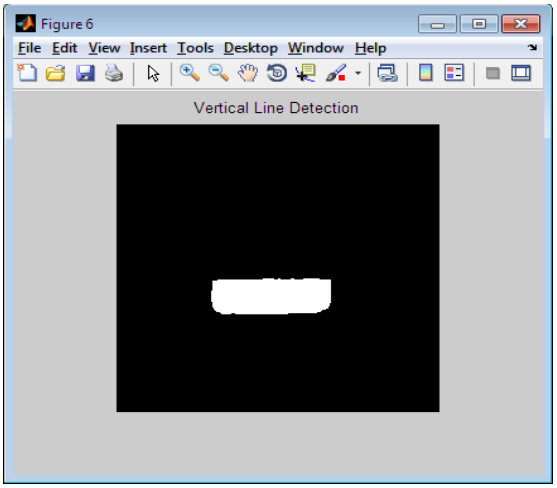
(B)



(C)

Fig 5.6. (A), (B), (C)Vertical Line Detection

After the unwanted lines are eliminated we have to press the vertical line detection button then we get the fig 5.6



(A)

After the vertical line detection we have to press the plate detection button then we will get the number plate on the original image shown in fig 5.7

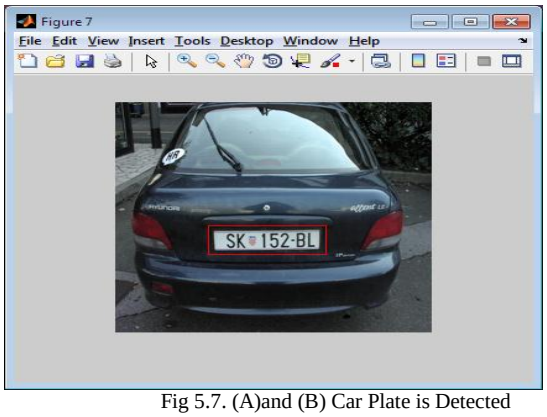


Fig 5.7. (A)and (B) Car Plate is Detected

VI. CONCLUSION

In this paper work a new and fast algorithm is proposed for vertical edge detection, in which its performance is faster .The VEDA contributes to make the whole proposed CLPD method faster.In proposed system a CLPD method in which data set was captured by using a web camera. Some set of images taken from various scenes and under different conditions.one LP is considered in each sample for the whole experiment.Finally the LP is detected accurately using the VEDA. Related to proposed methods, further study for improvement of the accuracy and practicality of this method is required. Also, the license plate detection is our main future work.

REFERENCES

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

[2] S. Thanongsak and C. Kosin, "*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

[3] H. Bai and C. Liu, "*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.

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

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

[6] J.-W. Hsieh, S.-H. Yu, and Y. S. Chen, "*Morphology-based license plate detection from complex scenes,"* in Proc. 16th Int. Conf. Pattern Recognit., Quebec City, QC, Canada, 2002, pp. 176–179.

[7].Mei Yu and Yong Deak Kim Ajou ,"*An Approach to Korean License Plate Recognition Based on Vertical Edge Matching*",in IEEE Xpl.,2000,pp.2975-2980.

[8]Abbas M. Al-Ghaili, Syamsiah Mashohor, Abdul Rahman Ramli, and Alyani Ismail, "*Vertical-Edge-Based Car-License-Plate Detection*," in IEEE Tras. On.Veh.Tec.,Vol 62, 2013, pp. 26–38.

