Automatic Vehicle Identification Using License Plate Recognition for Indian Vehicles

Sandra Sivanandan
Department of Computer Engineering
K. K. Wagh Institute Of Engineering Education & Research, Hirabai Haridas Vidyanagari
Amrut-Dham, Panchavati, Nashik-422003
University of Pune, Maharashtra

Ashwini Dhanait
Department of Computer Engineering
K. K. Wagh Institute Of Engineering Education & Research, Hirabai Haridas Vidyanagari
Amrut-Dham, Panchavati, Nashik-422003
University of Pune, Maharashtra

Yogita Dhepale
Department of Computer Engineering
K. K. Wagh Institute Of Engineering Education & Research, Hirabai Haridas Vidyanagari
Amrut-Dham, Panchavati, Nashik-422003.

Yasmin Saiyyad
Department of Computer Engineering
K. K. Wagh Institute Of Engineering Education & Research, Hirabai Haridas Vidyanagari
Amrut-Dham, Panchavati, Nashik-422003.

ABSTRACT
In this study, a smart and simple algorithm is presented for vehicle's license plate recognition system. The proposed algorithm consists of three major parts: Extraction of plate region, segmentation of characters and recognition of plate characters. For extracting the plate region edge detection and morphological operations are used. In segmentation part scan line algorithm is used. Character Segmentation for Devanagari Number Plates is also presented. Optical character recognition technique is used for the character recognition. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate.

Keywords
Devanagari, Edge detection, License plate recognition, Optical character recognition, segmentation.

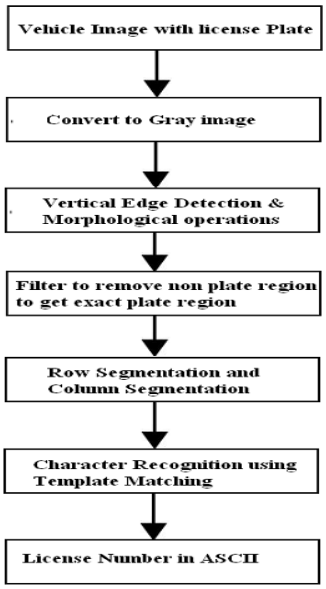
1. INTRODUCTION

License plate recognition (LPR) is a form of Automatic Vehicle Identification. It is an image processing technology used to identify vehicles by only their license plates. Real time LPR plays a major role in automatic monitoring of traffic rules and maintaining law enforcement on public roads. The LPR system's significant advantage is that the system can keep an image record of the vehicle which is useful in order to fight crime and fraud ("an image is worth a thousand words"). Early LPR systems suffered from a low recognition rate, lower than required by practical systems. The external effects (sun and headlights, bad plates, wide number of plate types) and the limited level of the recognition software and vision hardware yielded low quality systems. However, recent improvements in the software and hardware have made the LPR systems much more reliable and wide spread.

Here we are presenting a smart and simple algorithm for vehicle's license plate recognition system for Indian Vehicles. In this study, the proposed algorithm is based on extraction of plate region, segmentation of plate characters and recognition of characters. In India we find plates having Devanagari fonts as well (though according to rules it is not allowed). Character extraction for Devanagari font is slightly different as compared to English font because of the header line (shirorekha). We propose algorithm for character extraction for Devanagari font. The recognized plate can be then compared with police hotlist database to identify stolen vehicles.

The paper is organized as follows: Section II provides an overview of the overall system. Extracting the plate region is explained in Section III. Section IV gives the segmentation of individual plate characters. Section V deals with recognition of characters using optical character recognition based on statistical based template matching algorithm which uses correlation and section VI deals with verification of plate according to Indian rules. The paper concludes with Section VII.

2. STRUCTURE OF LPR SYSTEM



Flowchart of Proposed System

The algorithm proposed in this paper is designed to recognize license plates of vehicles automatically. Input of the system is the image of a vehicle captured by a camera. The captured image taken from 3-5 meters away is first converted to gray scale. We apply vertical edge detection algorithm and morphological operation i.e. open and close for plate extraction. After applying morphological operations image is filtered out to get exact plate region. Plate region is cropped. Row segmentation separates row in plate and column separation separates characters from row. Finally recognition part OCR recognizes the characters giving the result as the plate number in ASCII format. The result in ASCII format is can be verified on the basis of rules followed in India.

3. EXTRACTION OF PLATE REGION

Plate Extraction is done in following steps

3.1 Convert image to Gray Scale
3.2 Apply Vertical Edge detection
3.3 Candidate Plate Area Detection
- Morphologically Close image
- Fill holes in image
- Morphologically Open image

3.3 Filtration of non Plate region

3.1 Conversion To Gray Scale

This is pre-processing step for plate extraction. We apply Formula:

I(i, j) = 0.114*A(i, j,1) + 0.587*A(i, j, 2) + 0.299* A(i, j,3) where, *I(i,j)* is the array of gray image, *A(i,j,1), A(i,j,2), A(i,j,3)* are the R,G,B value of original image respectively. Sometimes the image may be too dark, contain blur, thereby making the task of extracting the license plate difficult. In order to recognize the license plate even in night condition, contrast enhancement is important before further processing [1].

Fig. 1) Original Image

Fig. 2) Gray Scale Image

Fig. 3) Gray image after contrast enhancement

3.2 Vertical Edge Detection

Before applying edge detection median filter is to be applied to image for removing noise. The main idea of median filter is to run through the signal, entry by entry, replacing each entry with the median of neighboring entries. Such noise reduction is a typical pre-processing step to improve the results of later processing (edge detection) [2].

unfiltered values		
6	2	0
3	97	4
19	3	10

median filtered		
*	*	*
*	4	*
*	*	*

In ascending order of values: 0, 2, 3, 3, **4**, 6, 10, 15, 97.
Center value (previously 97) is replaced by the median of all nine values (4).

Edge detection is performed on the given image, which aims at identifying points in digital image at which image brightness changes sharply or, more formally, has discontinuities. There mainly exists several edge detection methods (Sobel, Prewitt, Roberts, Canny).We use here Sobel operator for vertical edge detection.

If we define **A** as the source image, and **G***x* and **G***y* are two images which at each point contain the horizontal and vertical derivative approximations, the computations are as follows:

$$\mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A}$$

and

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A}$$

Where * is 2D convolution operation.

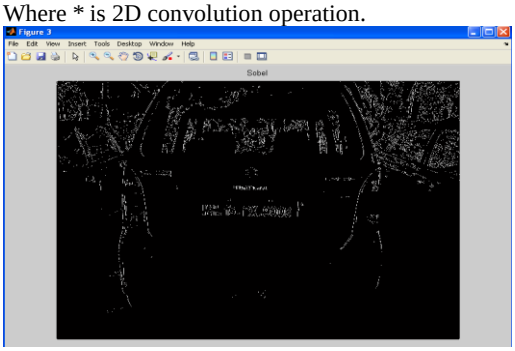


Fig. 4) Sobel Vertical Edge detection

3.3 Candidate Plate Area Detection

A morphological operator is applied to the image for specifying the plate location. We build a morphological operator that is sensitive to a specific shape in the input image. In our system rectangular box is employed as a structural element to detect the car plates.

In mathematical morphology structuring element are represented as matrices. Structuring element is a characteristic of certain structure and features to measure the shape of an image and is used to carry out other image processing operations [4].Typical rectangular structuring element is shown in figure.

1	1	1	1	1	1
1	1	1	1	1	1

A 2*5 'Rectangular' Structuring Element

Using two basic operation of morphology (erosion and dilation), opening and closing of image is done. The opening of *A* by *B* is obtained by the erosion of *A* by *B*, followed by dilation of the resulting image by *B*. The closing of *A* by *B* is obtained by the dilation of *A* by *B*, followed by erosion of the resulting structure by *B*.

For closing image 10*20 rectangular structuring element is used. After closing image we have to fill the holes in this image. A hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image [3]. Then image is opened using 5*10 rectangular structural element. Values are determined according to the size of the image. Here we have used 1280X980 resolution images.

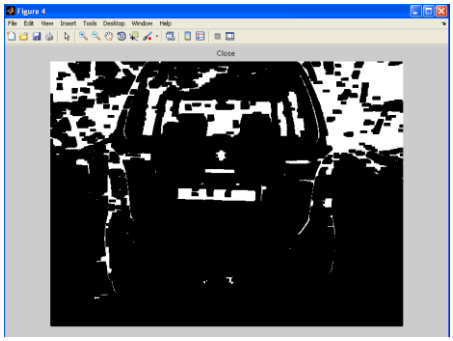


Fig. 5) Closed Image

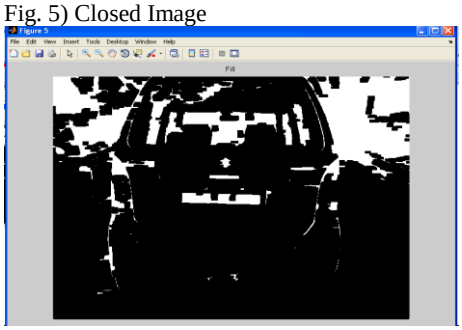


Fig. 6) Filled Image

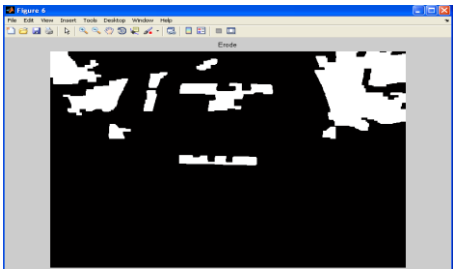


Fig. 7) Opened Image

3.4 Filtration Of Non Plate Region

After identify the ROI, image is then filtered using following filtering techniques. First find the connected components in image.

The first technique involves removing of all white patches which has more or less area than the threshold. For instance components having area < 2000 or >20000 are eliminated.

Using Bounding Box method, draw Bounding Box around components and fill the image.

According to the height values, for instance, only the objects with a height greater than *T*min_*h* and less than *T*max_*h* are retained, and eliminate the other objects. After that, if the width values of the retained objects are greater than *T*min_*w* and less than *T*max_*w*, the objects are retained; otherwise, the objects are removed, and so on. Where:

*T*min_*h* : Minimum height of the object.
*T*max_*h* : Maximum height of the object.
*T*min_*w* : Minimum width of the object.
*T*max_*w* : Maximum width of the object [6].

After filtering plate region is cropped by searching for the first and last white pixels starting from top left corner of an image. Plate is cropped from original image after getting coordinates.

Fig 6) Filtered Image on basis of area

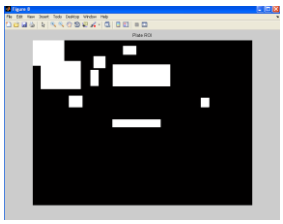


Fig. 7) Bounding Box and filled image

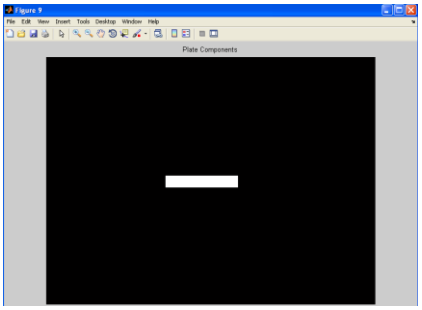


Fig. 8) Image after filtration on basis of height &width of objects

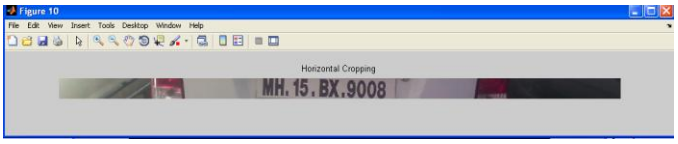


Fig. 9a) After Horizontal Cropping

Fig. 9b) After Vertical Cropping: Final Plate image

If the plate is not straight then characters will not be extracted properly. So the Plate image must be rotated to make it straight. For tilt correction we use orientation property of connected component. Orientation Property returns angle through which plate is rotated in opposite direction. e.g. If angle returned by Orientation Property is 5 degree, then Plate image has to be rotated by -5 degree.

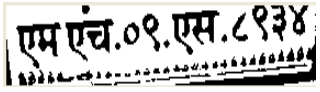



Fig. 9c) Tilted Plate


Fig. 9e) After Tilt correction

4. SEGMENTATION OF PLATE CHARACTERS

Before applying the OCR, the individual lines in the text are separated using line separation process and individual characters from separated lines.

Steps for Character Segmentation:

4.1 Binarization of Plate image
4.2 Scan Line Algorithm for row segmentation
4.3 Vertical Projection for column segmentation

4.1 Binarization Of Plate Image

Binarize the plate image. Threshold for binarization must be such that characters are displayed well. For that we take average of all pixel values in plate image and calculate threshold.

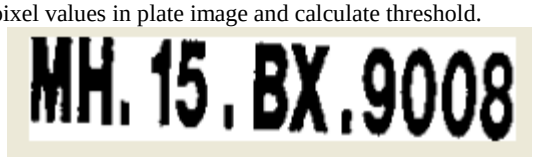


Fig. 10) Binarized image

4.2 Scan Line Algorithm

The scan line algorithm is based on the feature that there is transition from 1 to 0 and 0 to 1 transition in character region in a binary image. Thus the total number of transition in character region is more than the total number of transition in other region. There are at least seven characters in license plate region and every character has more than two Jumps[7]. We can choose twelve as the threshold value. If the total number of transitions in a certain line is greater than twelve, this line may be in character region. Otherwise, it is not in character region.

Algorithm:

1) Let H be height and W be Width of Plate image.
2) for(i=H/2 to 0)
   {
   Count no of transitions ie 0 to 1 and 1 to 0 in cnt;
   if cnt <12 get y coordinate in ymin and break;
   }
3) for(i=H/2+1 to H-1)
   {
   Count no of transitions ie 0 to 1 and 1 to 0 in cnt;
   if cnt <12 get y coordinate in ymax and break;
   }
4) crop the image from ymin to ymax.

Thus we get Top and bottom boundary of Plate. Now we can segment the characters.

Fig. 11) After Scan Line Algorithm

4.3 Vertical Projection

Scan the cropped image from left to right column by column after precise location of the top and bottom boundary and count the total number of black points in every column.
The threshold value is set to h/10. Judge every value in array projection. If Projection[i] is greater than h/10, Projection[i] is set to one. Otherwise, Projection[i] is set to zero. Where h is the modified no of rows of the binary image after precise location of top and bottom boundaries [7].

Then characters are cut by selecting the portions having Projection[i]=1. Before recognition algorithm, the characters are to be refined into a block containing no extra white spaces (pixels) in all the four sides of the characters.

Fig. 12 a) Character Segmentation for English font

Fig 12 b) Characters after removing extra spaces from four sides

CHARACTER SEGMENTATION FOR DEVANAGARI [9]

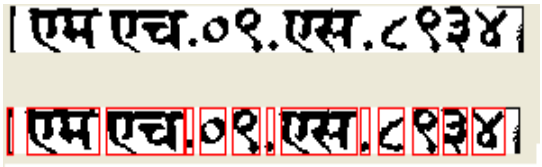


Fig. 12 c) Character Segmentation for Devanagari font

As shown in figure 12 c) Devanagari Numbers can be separated by using vertical projection but due to header line we cannot separate single character from the word. So we present here algorithm for extraction of Devenagari character.

Fig 12 d) Horizontal Projection of Devanagari word and header width.

Fig 12 e) After Removing Header Line

Fig 12 f) Character Extraction for Devanagari font

ALGORITHM

1) Find the Header line by finding Horizontal Projection of the Word i.e. by finding rows having maximum black pixels.
2) Find the Header Width.
3) Remove the Header line. After removing header line our word is divided into three horizontal parts known as upper zone, middle zone and lower zone.
4) Apply Vertical Projection after removing header line.
5) Extract the characters as in previous algorithm.

5. RECOGNITION OF CHARACTERS

The OCR is now used to compare the each individual character against the complete alphanumeric database using template matching. Template matching is one of the Optical Character Recognition techniques [8]. The image is converted into 12x12 bitmap. Bitmap is represented by 12x12-matrix or by 144 vectors with 0 and 1 coordinates.

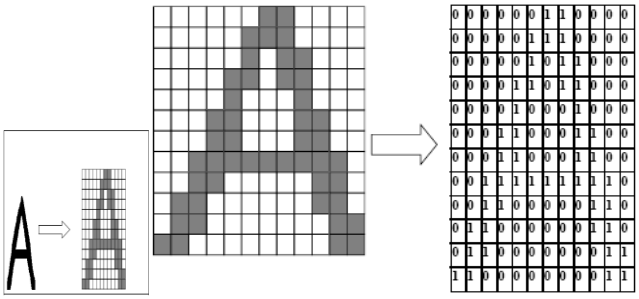


Fig. 13) Bitmap represented by 12x12 matrix

The character image is compared with the ones in the database and the best similarity is measured. The OCR actually uses correlation method to match individual character.

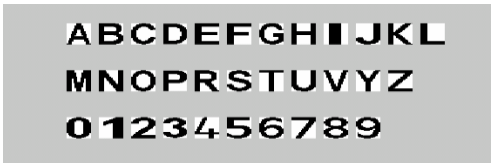


Fig.14) Database characters

This process involves the use of a database of characters or templates. There exists a template for all possible input characters. For recognition to occur, the current input character is compared to each template to find either an exact match, or the template with the closest representation of the input character. If *I(x, y)* is the input character, *Tn(x, y)* is the template n, then the matching function *s(I, Tn)* will return a value indicating how well template *n* matches the input character .Some of the more common matching functions is correlation based on the following formula:

$$s(I,Tn) = \frac{\sum_{i=0}^{w}\sum_{j=0}^{h}(I(i,j)-|I|)(Tn(i,j)-|Tn|)}{\sqrt{\sum_{i=0}^{w}\sum_{j=0}^{h}(I(i,j)-|I|)^2}\sqrt{\sum_{i=0}^{w}\sum_{j=0}^{h}(Tn(i,j)-|Tn|)^2}}$$

6. VERIFICATION OF PLATE

The plate number thus recognized can be stored in an array and can be verified on the basis of rules followed in India [4]. In the case of Indian number plates, the length can be 8, 9 or10. Rules corresponding to the number plate lengths are as follows:

- If the number of segmented characters counts to 7, the number plate follows 1939 series having first three characters as alphabets and rest as numbers.
- If the number of character counts to 8, the first two characters should be alphabets and rest should be numbers.
- If the number of character counts to 9, the first, second and fifth characters should be alphabets and rest should be numbers.
- If the number of segments counts to 10, the first, second, fifth and sixth characters will be alphabets and rest will be numbers.

7. CONCLUSION

In this paper, we presented application software designed for the recognition of car license plate. Firstly we extracted the plate location, and then we separated the plate characters individually by segmentation and finally applied template matching with the use of correlation for recognition of plate characters. This system is designed for the identification Indian license plates and the system is tested over a small number of images. We have also proposed algorithm for extraction of Devanagari characters. In future studies we will propose algorithm for different regional languages plate character segmentation and recognition.

8. ACKNOWLEDGEMENT

We are greatly indebted to **Prof. Miss Neha Sharma,** Department of Computer Engineering, K.K.W.I.E.E.R, Nashik, our internal guide for her guidance & enlightening comments throughout the project work. It has been an altogether different experience to work with her & we would like to thank her for her helpful suggestion & numerous discussions.

We gladly take this opportunity to thank **Prof. Dr. S. S. Sane** [Head of Dept., Computer Engineering] for providing facilities during progress of the project.

.

9. REFERENCES

[1] Chirag N. Paunwala & Suprava Patnaik,"A Novel Multiple License Plate Extraction Technique for Complex Background in Indian Traffic Conditions", International Journal of Image Processing (IJIP) Volume (4): Issue (2).

[2] Ayesha Butalia, Durgesh Maru, Kuldeep Baheti, Avais Mohammad, Ankit Bagdiya, "Gray Eye Traffic Surveillance", International Journal of Computer Trends and Technology- May to June Issue 2011.

[3] Chetan Sharma1 and Amandeep Kaur ,"Indian Vehicle License Plate Extraction And Segmentation", International Journal of Computer Science and Communication

[4] Phalgun Pandya, Mandeep Singh, "Morphology Based Approach To Recognize Number Plates in India", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-3, July 2011.

[5] Muhammad Tahir Qadri, Muhammad Asif, "Automatic Number Plate Recognition System For Vehicle Identification Using Optical Character Recognition", 2009 International Conference on Education Technology and Computer.

[6] Kumar Parasuraman, and P.Vasantha Kumar "An Efficient Method for Indian Vehicle License Plate Extraction and Character Segmentation", 2010 IEEE International Conference on Computational Intelligence and Computing Research

[7] Ch.Jaya Lakshmi, Dr.A.Jhansi Rani, Dr.K.Sri Ramakrishna, M.KantiKiran, "A Novel Approach for Indian License Plate Recognition System", International Journal Of Advanced Engineering Sciences And Technologies Vol No. 6, Issue No. 1, 010 - 014

[8] Nadira Muda, Nik Kamariah Nik Ismail, Siti Azami Abu Bakar, Jasni Mohamad ZainFakulti Sistem Komputer & Kejuruteraan Perisian, "Optical Character Recognition By Using Template Matching (Alphabet)".

[9] Raghuraj Singh, C. S. Yadav, Prabhat Verma, Vibhash Yadav, "Optical Character Recognition (OCR) for Printed Devnagari Script Using Artificial Neural Network", International Journal of Computer Science & Communication, January-June 2010