A Study on Vehicle Number Plate Identification by Morphological Edge Detection and Template Matching

Dipankar Bhattacharya and Anjan Bikash Maity

Abstract—**Vehicle License Plate Recognition is an important area of Image processing which has various applications. In situations where no standard number plate format and font is maintained, identification becomes difficult. In this paper, we discussed effect of Edge Detection and Template Matching on identification of Vehicle Number Plates with various types of fonts. Experimental results show that the performance of the proposed method is simple and satisfactory, considering the complexity of the input image.**

Indexed Terms—**Character Recognition, Edge Detection, Template Matching.**

I. Introduction

Vehicle License Plate Recognition (LPR) has become a significant application in the transportation system. It can be used in many applications such as entrance admission, security, parking control, and road traffic control, and speed control. However, when no standard font style and size is maintained in different Vehicle Number Plates, identification of characters in a Number Plate becomes difficult .Detection of location of the Number Plate from image is also difficult task. All these problems restrict the use of automated Vehicle Number Plates Identification System in various complex situations and places. The basic idea behind our work is to use morphological edge detection method and simple character recognition technique in a complex scenario to find if the results are satisfactory. Template matching is regarded as a common methodology for identification of characters from an image. When the characters are of predefined font this produces very satisfactory result. However in complex situation like Vehicle License Plate Recognition where the images are noisy and fonts may not be of a same type, applying this methodology and obtaining a significant successful outcome is challenging. In our proposed technique we have experimented on all these possibilities and displayed the results, which are satisfactory.

This paper is organized as follows. The proposed model and steps of the system are explained in section 2. Experimental results and analysis are presented in section 3.The Conclusion is summarized in section 4.

II. Proposed Model

Our proposed model consists of Digitization of image, Edge Detection, Separation of character, Template matching and Experimental Result.

A. Digitization

The process of digitization is important for input image used in the system. In this process, the input image is sampled into a binary window, which forms the input to the recognition system. In the above figure, the alphabet A has been digitized into digital cells, each having a single color, either black or white. It becomes important for us to encode this information in a form meaningful to a computer. For this, we assign a value +1 to each black pixel and 0 to each white pixel and create the binary image matrix, which is shown in the Fig. A . Digitization of an image into a binary matrix of specified dimensions makes the input image invariant of its actual dimensions. Hence an image of whatever size gets transformed into a binary matrix of fixed pre-determined dimensions. This establishes uniformity in the dimensions of the input and stored patterns as they move through the recognition system.

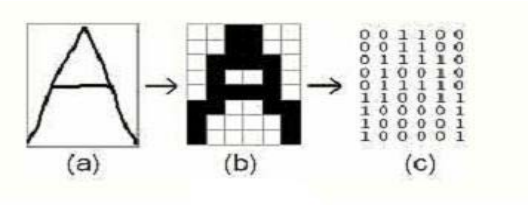

Fig A

B. Morphological Edge Detection

Then we follow the following four steps.

- Boundary extraction,
- Hit-or-miss transform
- Thinning operation
- Pruning operation

1) Boundary Extraction

Our first step for morphological edge detection is boundary extraction. At first we take a structure element and make erosion on the image by this structure element. Then we erode the binary image.

The erosion may be defined as:

$$A \Theta B = \text{Minimum}\,(A(x-i, y-j) \times B(i, j)) \quad (1)$$

This expression shows that we make the erosion of image A by the structure element B. The structure element is entirely contained within A. In fact, erosion reduces the number of pixels from the object boundary. The number of pixels removed depends on the size of structure element.

Manuscript received July 27, 2011; revised August 10, 2011.

Dipankar Bhattacharya, Asst Prof ,Dept Of Information Technology, Calcutta Institute Of Technology, dipankar_bha@yahoo.com

Anjan Bikash Maity, Lecturer ,Dept Of Information Technology, Calcutta Institute Of Technology, maity.anjan@gmail.com

Let us consider
A = {(1,0),(1,1),(1,2),(0,3),(1,3),(2,3),(3,3), (1,4)}
B = {(0, 0), (1, 0)}
A Θ B = {(0, 3), (1, 3), (2, 3)}

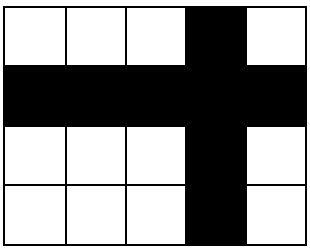

Fig B.1.1: Original image

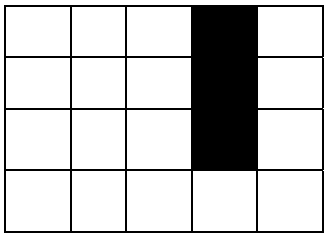

Fig B.1.2:Eroted Image

After eroding we subtract this eroded image from the binary image. Then we got a boundary extracted image, which is two or more pixel thick noiseless binary image. The boundary extraction may be defined as:

$$C = A - (A \Theta B) \qquad (2)$$

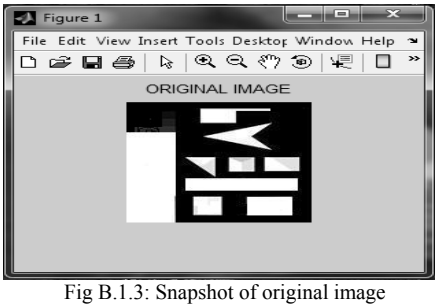

Fig B.1.3: Snapshot of original image

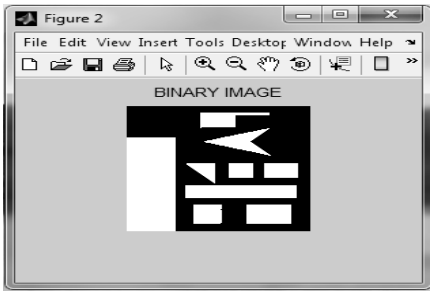

Fig.B.1.4: Snapshot of binary image

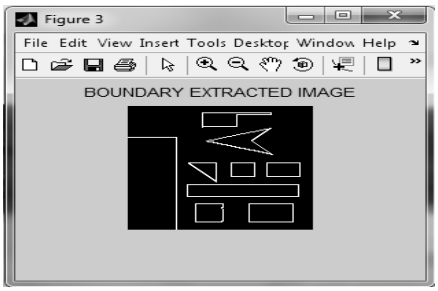

Fig.B.1.5: Snapshot of boundary extract image

2) Hit-Or-Miss Transform

The hit-or-miss transformation may be defined as morphological operator, which is used for making one pixel thick image from two or more pixel thick image. A small odd size mask typically, 3 × 3, can be scanned over a binary image. The hit-and-miss transformation operates as a binary matching between the image and the structuring element. If the foreground and background pixels in the structuring element exactly match the foreground and background pixels in the image, then the pixel underneath the origin of the structuring element is set to the foreground color. If it does not, that pixel is set to background color. The Hit-or-Miss transform may also be expressed in terms of erosion as:

$$A \otimes B = (A \Theta B_1) \cap (A^c \Theta B_2) \qquad (3)$$

In this process, at first we make the Erosion Operation on the image A with the structure element B1. Then calculate the complement of image A. Then, again we make the Erosion Operation on the image which is the complement of A with the structure element B2. At last we make the Intersection Operation between the two eroded image and we find the result of the Hit-or-Miss Transformation

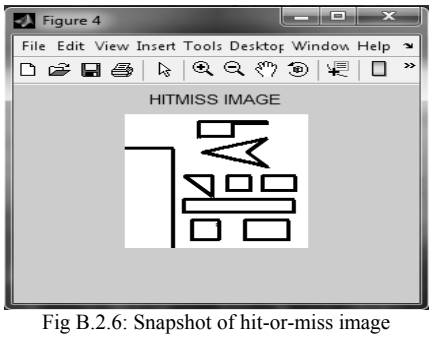

Fig B.2.6: Snapshot of hit-or-miss image

3) Thinning Operation

Thinning is a morphological operation that is used to remove selected foreground pixels from binary images, somewhat like erosion or. It can be used for several applications, but is particularly useful for skeletonization. In this mode it is commonly used to tidy up the output of edge detectors by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images, and produces another binary image as output. The thinning operation is related to the hit-or-miss transform.

Thinning Operation may be defined by the following expression:

$$A \text{ Ø } B = A - (A \otimes B) \qquad \ldots\ldots \quad (4)$$

In this operation we must be subtract the Hit-or-Miss Transformed image from the Boundary extracted image.

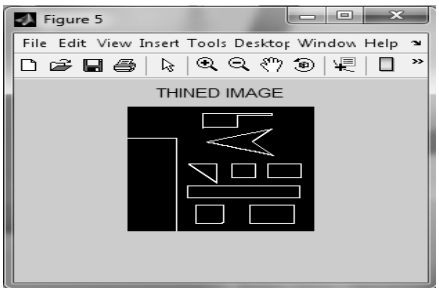

Fig.B.2.7: Snapshot of thinning image

4) Pruning Operation

Thinning operation may be reduced the thickness of an object in an image to a one pixel wide skeleton representation. The problem with this operation is that they leave behind extra tail pixels.

The tail pixels required to remove from this image. The process of removing these tail pixels is called as Pruning. In morphological pruning operation we use the hit-and miss operation with this image by a composite structuring element. After the pruning operation we got the original edge of an object in an image. The morphological operation may be defined as:

$$C = A \otimes B \quad (5)$$

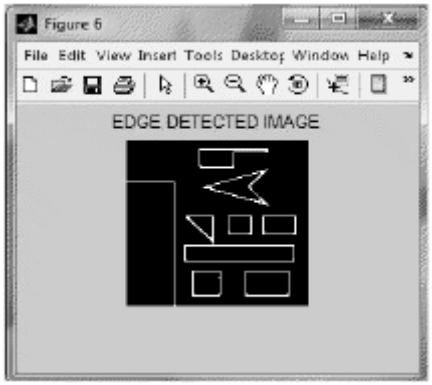

Fig.B.2.8: Snapshot of edge detected image

C. Corner Detection

Corner detection is an important aspect in image processing and researchers find many practical applications in it. There are many algorithm for detect corner like, Corner detector based on global and local curvature properties. But, we use the Freeman chain codes algorithm for detect true corner of an object in an image.

1) Chain Codes Algorithm

Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. Typically, this representation is based on 4- or 8-connectivity of the segments. The direction of each segment is coded by using a numbering scheme such as ones

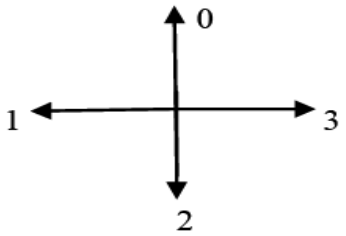

Fig. C.1: 4-directional chain code

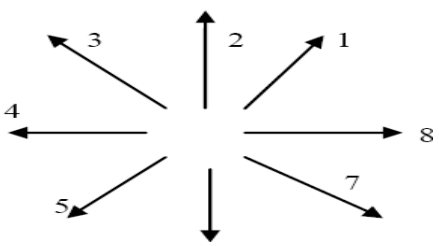

Fig. C.2: 8-directional chain code

The chain code of a boundary depends on the starting point. However, the code can be normalized with respect to the starting point by treating it as a circular sequence of direction numbers and redefining the starting point so that the resulting sequence of numbers forms an integer minimum magnitude. We can normalize for rotation by using the difference of the chain code instead of the code itself. The difference is obtained by counting the number of direction changes that separate two adjacent element of the code.

Another advantage of chain code is that it is translation invariant but not as well for rotation and scaling. Rotation can be resolved by taking difference chain code while scaling by addressed by changing the size of the sampling grid which the shape overlays on. Each segment of the chain code has direction. If we want to move from one segment to another, the angle magnitude will change.

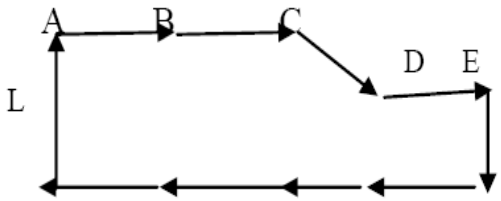

Fig.C.3: Change direction in 90 and 45 degree

For example, from the above figure if we start from point A, then the chain code for this figure is 00706444422 in 8-directional. Here, From point A to point B the chain code is 0 and point B to C again 0. Here the code will not change so here corner will not be detected. Point C to point D the chain code is 7, here the chain code is change so in point C a corner will be detected. Point D to point E the chain code is 0, here the chain code again change so in point D again a corner will be detected

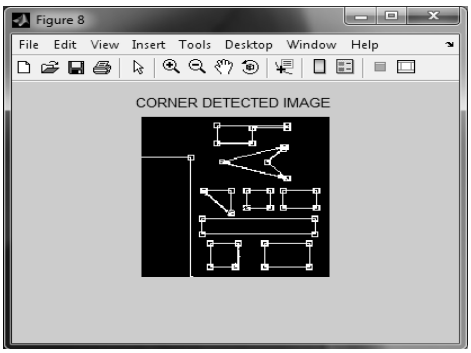

Fig.C.4: Snapshot of corner detected image

D. Separation of Character

After identifying the corner locations we extract the Vehicle Number Plate image section from the original image. Separation of character has been done in two phases, namely, Horizontal Segmentation and Vertical segmentation. At first, the image is processed row-wise to separate useful information. After this phase we have been able to identify the series of characters sequentially arranged from a Number plate image. Following Vertical segmentation, which is column-wise operation to separate information, we finally

have characters totally separated from an input Number Plate image. This process is explained in Figure D.

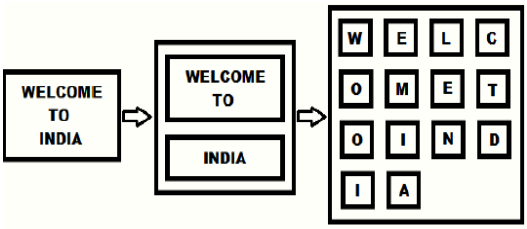

Fig D

E. Template Matching

In this step, template matching by correlation method is applied on the separated characters for identification. A correlation is a single number that describes the degree of relationship between two variables. The quantity *r*, called the *linear correlation* coefficient, measures the strength and the direction of a linear relationship between two variables which is described as

$$r = \frac{n\sum xy - \left(\sum x\right)\left(\sum y\right)}{\sqrt{n\sum x^2 - \left(\sum x\right)^2} \cdot \sqrt{n\sum y^2 - \left(\sum y\right)^2}}$$

where *n* is the number of pairs of data.

The value of r is such that $-1 \leq r \leq +1$. The + and – signs are used for positive linear correlations and negative linear correlations, respectively. Threshold value is .1000.

In our proposed model, pixel values of template characters (A-Z, 0-9) of size 42x24 are stored in vector such that vector location 1 stores value for character A, location 2 for B and so on. The identification process done as follows

1. Get the value of CORRELATION COEFFICIENT (R) between template character matrix and input character matrix.
2. Find the maximum value of (R) and corresponding template matrix location.
3. Find the character corresponding to that location and produce output.

The processes explained in Fig E.

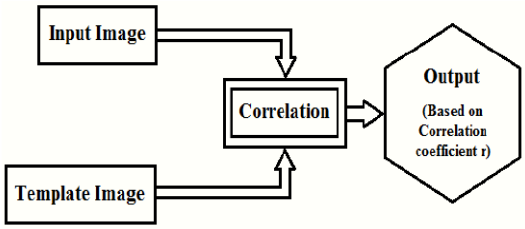

Fig E

III. EXPERIMENTAL RESULTS

From the show Results in MATLAB environment it is evident that the system is capable of producing computer readable data from image containing text. Points to be considered that, in this paper only 1 template set is compared (which consists of characters of a particular font) with different font characters. In different cases fonts of character is different from each other but when template matching is done on those input images, the out put is quite accurate irrespective of the fonts styles in image. The images contain noises, which have been removed during processing as pixel below a particular size was discarded.

Fig F Edge Detection of the Number Plate

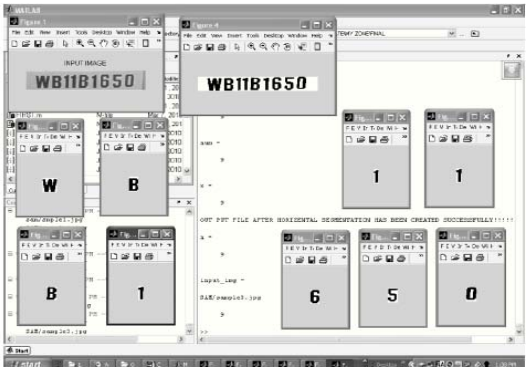

Fig G Character Separation

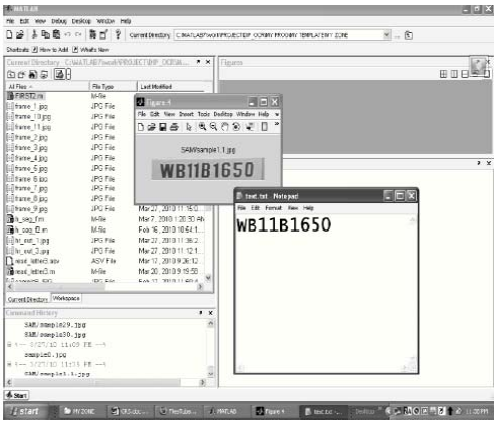

Fig H Character Identification

TABLE 1

NO OF CAR PLATES	NUMBER OF CHARACTERS	IMAGE QUALITY	ACCURACY
10	90	Poor	66%
10	90	Average	79%
10	90	Good	93%

Analysis of results

IV. CONCLUSION

In this paper we overviewed the problem of Vehicle Number Plate recognition with various types of Number Plates. We proposed a correlation based character recognition system, which provides result with significant accuracy, which is very simple to implement. The system has been tested on MATLAB environment with satisfactory

results. Most of the time the input image taken from low-resolution mobile camera which does not have very good quality image output. Given a better device the result should increase in accuracy significantly.

V. ACKNOWLEDGMENT

Authors of this paper sincerely thank all the members of Information Technology Department of Calcutta Institute of Technology for their constant support and encouragement. Authors also sincerely thank the Institution Authorities for their support.

REFERENCES

[1] Fu, K.S.and Rosenfeld, A. "Pattern Recognition and Computer Vision", Computer , vol.17, no.10, pp.274-282, Oct. 1984.Fu**,** K.S., and Young, T**.**, (Eds.), Handbook of Press1986 Pattern

[2] Amitava Sen,Suman Bhowmik : A New Parallel Algorithm for Identification of Straight Lines in Images, Journal of parallel computing,2005.

[3] Arijit Laha Amitava Sen and Bhabani P. Sinha : Parallel Algorithms for Identifying Convex and Non-convex Basis Polygons in an Image. Journal of parallel computing,2009.

[4] "Digital Image Processing", Gongalez.

[5] M.A. Ko and Y.M. Kim, "License Plate Surveillance System Using Weighted Template Matching", Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop, October 2003, pp.269-274.

[6] Neeta Nain, Viajay Laxmi, Ankur Kumar Jain and Rakesh Agarwal. "Morphological Edge Detection and Corner Detection Algorithm Using Chain-Encoding**"**, IPCV'2006, pp 1-5.

Dipankar Bhattacharya, is an Asst Prof in Calcutta Institute Of Technology, Uluberia,Howrah,West Bengal,India.He has done his B.Tech from Narula Institute Of Technology and M.Tech from BESU,Shivpur. He has more than 3 years of teaching experience at B.Tech level. His interests include Image processing and Web Applications

Anjan Bikash Maity, is a faculty member in Calcutta Institute Of Technology, Uluberia,Howrah,West Bengal,India.He has done his B.Tech from Bankura Unnayani Institute Of Engineering and M.Tech from Haldia Institute of Technology. He has more than 2 years of teaching experience at B.Tech level..His interests include Image processing and Cryptography.