IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Special Issue-1 E-ISSN: 2347-2693

Automatic Number Plate Recognition (ANPR) of Vehicle using Image processing and Graph based Pattern Matching

Bibek Ranjan Ghosh[1*], **Siddhartha Banerjee**[2]
Dept. of Computer Science
Ramakrishna Mission Residential College,
Narendrapur , Kolkata -103
[1]bibekghosh2003@yahoo.co.in
[2]sidd_01_02@yahoo.com

Prithviraj Pramanik[3], **Soham Dey**[4]
Dept. of Computer Science
Ramakrishna Mission Residential College,
Narendrapur , Kolkata -103
[3]prithvirajpramanik@yahoo.co.in
[4] sohamgenius@gmail.com

Abstract — **Automatic Number Plate Recognition (ANPR) is a real time embedded system which identifies the characters directly from the image of the license plate. Since different standard are used in different country, automatic number plate recognition system is different for each country. In this paper, a number plate recognition system for vehicles in India is proposed. This paper introduces a vehicle number plate identification system, which extracts the characters features of a plate, from an image captured by a digital camera. The system uses broad steps like thresholding, image segmentation, thinning and pattern matching for extraction of characters. The pattern matching is done using graph based method. Graph matching techniques are introduced to compute the similarity of characters extracted from number plate with the information of characters store in the database.**

Keywords: APNR, Segmentation, Thinning, Scaling, Pattern Matching, COG, Complete Bipartite Graph, Adjacency Matrix

I. INTRODUCTION

A vehicle registration plate is a metal or plastic plate attached to a motor vehicle for official identification purposes. The registration identifier is a numeric or alphanumeric code that uniquely identifies the vehicle within the issuing region's database. In some countries, the identifier is unique within the entire country, while in others it is unique within a state or province. Whether the identifier is associated with a vehicle or a person also varies by issuing agency.

All motorized road vehicles are tagged with a registration or license number in India. The license plate (commonly known as number plates) number is issued by the district-level Regional Transport Office (RTO) of respective states — the main authority on road matters. The license plates are placed in the front and back of the vehicle. By law, all plates are required to be in modern Hindu-Arabic numerals with Latin letters. Other guidelines include having the plate lit up at night and the restriction of the fonts that could be used.

Plates for private car and two-wheeler owners have black lettering on a white background. Commercial vehicles such as taxis and trucks have a yellow background and black text. The President of India and state governors travel in official cars without license plates. Instead they have the Emblem of India in gold embossed on a red plate [1].

The current format of the registration index consists of 3 parts, they are:

- The first two letters indicate the state to which the vehicle is registered.
- The next two digit numbers are the sequential number of a district. Due to heavy volume of vehicle registration, the numbers were given to the RTO offices of registration as well.

- The third part is a 4 digit number unique to each plate. A letter(s) is prefixed when the 4 digit number runs out and then two letters and so on.

The purpose of this paper was to build a real time application which recognizes license plates from cars at a gate, for example at the entrance of a parking area. The system, based on regular PC with digital camera, catches the frames which include a visible car license plate and processes them. Once a license plate is detected, its digits are recognized by checking against a database.

Identification of vehicle license plate is basically a five step process [2]:

1) Image acquisition i.e. capturing the image of the license plate
2) Pre-processing the image i.e. normalization, adjusting the brightness, skewness, and contrast of the image
3) Localizing the license plate
4) Character segmentation i.e. locating and identifying the individual image on the plate, and
5) Pattern matching.

II. RELATED WORK

The problem of automatic VNP recognition is being studied since the 90's. The early approaches were based on characteristics of boundary lines. The input image being first processed to enrich and enhance boundary line information by using algorithms such as the gradient filter, and resulting in an image formed of edges. The image thus processed was converted to its binary counterpart and then processed by certain algorithms, such as Hough transform, to detect lines. Eventually couples of 2-parallel lines were considered as a plate-designate.

Pawan Wawage & Shraddha Oza [2] proposed a vehicle number plate identification system, which extracts the characters features of a plate from a captured image by a digital camera. This paper deals with computing techniques from the field of Artificial Intelligence, machine vision, and neural networks in construction of an Automatic System for Identification of Vehicle License Plate and Character Recognition. The system consists of the following standard modules: 1. Edge Detection, 2. Selection of probable Band, 3. Number Plate Localization, 4. Skew detection and de-skewing, 5. Character Segmentation, and 6. Character Recognition.

Othman Khalifa, Sheroz Khan, Rafiqul Islam, and Ahmad Suleiman [3] proposed a method to perform recognition of license plates under any environmental condition, with no assumptions about the orientation of the plate or its distance from the camera. To solve the problem of localization of a license plate, a simple texture-based approach based on edge information is used. Segmentation of characters is performed by using connected components analysis on the license plate's image and a simple multi-layer Perceptron neural network is used to recognize them. Simulation results were shown to be an efficient method for real time plate recognition.

D. Bharath Kumar, V. E. Chandrashekar [4] proposed a method for Segmentation of the plate characters was achieved by thresholding, labeling and filling up the holes approach was planned during this paper. Median filter was additionally applied to get rid of the unwanted noise that is suffering from the scanning device.

Ankush Roy, Debarshi Patanjali Ghoshal [5] proposed a method in which considerable focus was given on the segmentation algorithm that was based on tagging the pixel cluster and a region growing approach.

Today many commercial systems like the Smartreg [6], Car Plate Recognition by J.A.G. Nijhuis [7], Automatic Number Plate Recognition (ANPR) by Shyang-Lih Chang, Li Shein Chen, Yun-Chung Chung, and Sei-Wan Chen [8] are available.

All of the systems discussed above have some kind of limitations for example they are plate size dependent, work only in certain conditions or environment like indoor images etc. The method that we are proposing is independent of size, location and angle of the number plate of the vehicle.

III. PROPOSED METHOD

An overview of the proposed number-plate recognition system is given in Figure 1.

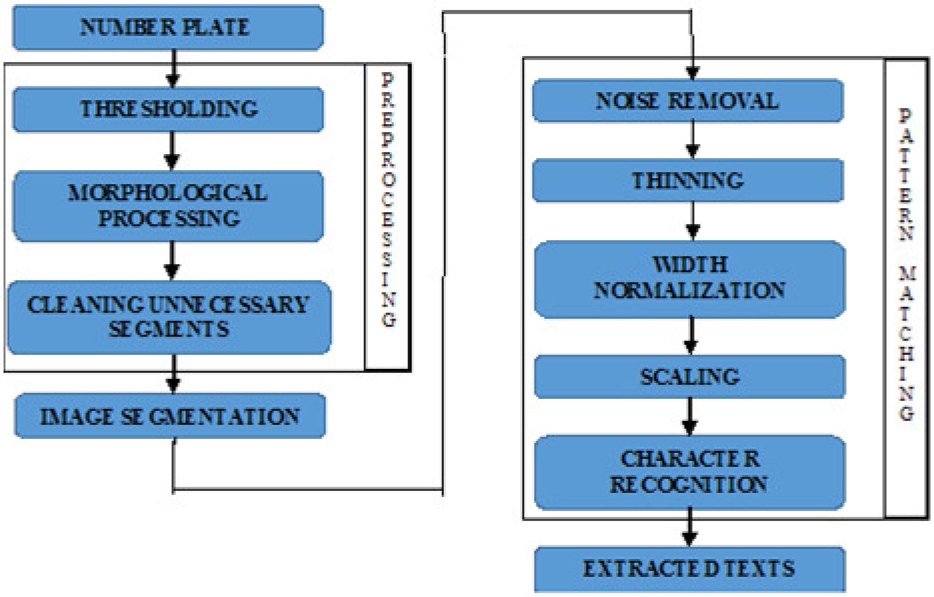

Figure 1: Steps of Automatic Number Plate Recognition System

A. Pre-processing

The image of the number plate captured by camera may contain some noise. To remove these noises present in the image and make it compatible for further processing we apply the following steps.

Thresholding: The gray-scale image of the number plate is binarized using intensity based thresholding. It should be noted that the number plate is captured by camera in gray-level. The Otsu [9] thresholding method is used to choose a threshold, *th*. B[i ,j] = 1 if F[i, j]>th and B[i ,j] = 0 other wise. F[i, j] and B[i ,j] denotes the pixel at [i, j] location of the gray scale image and the corresponding image after binarization.

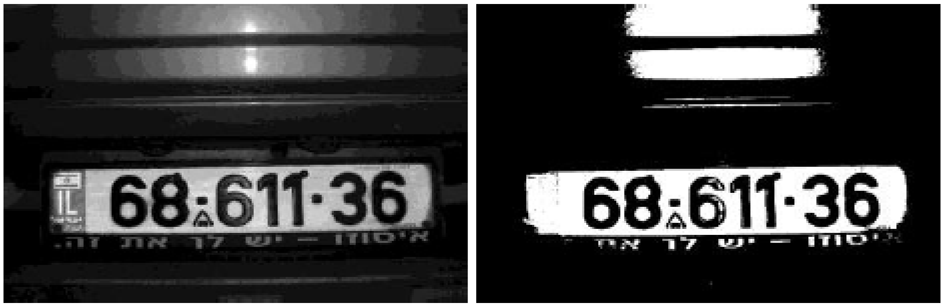

Figure 2: (a) original Number Plate (b) Threshold image

Morphological processing: After thresholding, to separate the characters from each other in the number plate an opening operation is done. Opening [10] operation consists of morphological erosion followed by dilation. Erosion with small square structuring elements shrinks an image by stripping away a layer of pixels from both the inner and outer boundaries of regions. The holes and gaps between different regions become larger, and small details are eliminated. In order to retain the original size of the characters of the number plate, dilation is (a) (b) carried out with the eroded image. In our experiment, 3× 3 structuring element has been considered for opening operation. Thus, the output after morphological operation on image in Figure 2(b) has been shown in Figure 3.

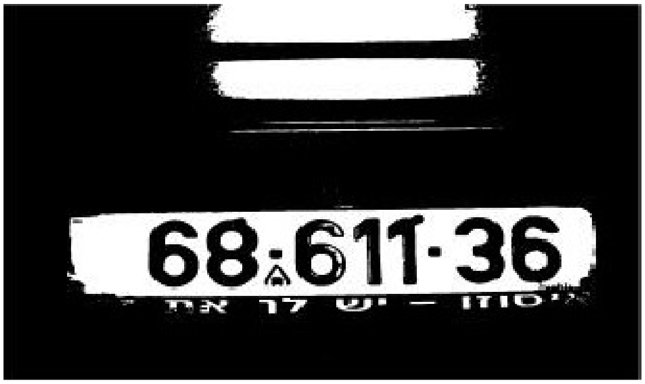

Figure 3: After Morphological Opening using 3X3 Structural Element

Cleaning Unnecessary Segments: To clear unnecessary segments other than characters, we first marked each region by a unique number by using region growing [10] algorithm. The algorithm is given below.

ALGORITHM 1
Step 1: An arbitrary black pixel (r, c) known as s*eed* pixel is chosen from the image.
Step 2: This pixel is given a unique label *L*.
Step 3: 8-neighbouring pixel of the seed pixel is examined. The neighbour(s) that are black are also assigned the unique number *L*. Once a new pixel is accepted as a member the neighbors of this new pixel are examined. This process goes on recursively until no more pixels is accepted. All the pixels of the current region are marked with *L*.
Step 4: Repeat Step 1 to Step 4 until every pixel is assigned to some region.
Next we calculate average number of pixels in each region. If a region R contains less number of pixels than the average we remove the region R. Figure 4 shows the number plate of Figure 3 after removing the unnecessary segments.

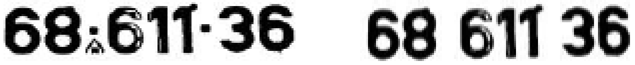

Figure 4: a) Before cleaning b) After cleaning

B. Image Segmentation

For Image segmentation we apply same region growing algorithm given in ALGORITHM 1 to level each black region with a level. Then we create separate image for each level. The application of this step on Figure 4 is shown in Figure 5.

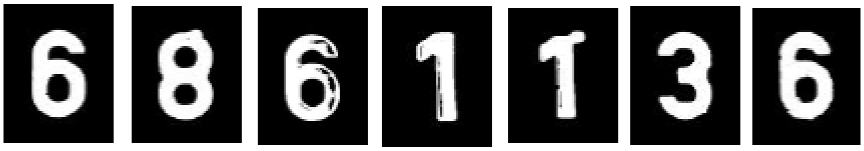

Figure 5: Segmentation of characters (image negative)

C. Pattern Matching

After segmentation of each character we follow a pattern matching step to compare the features of the extracted characters with the features of the character stored in the data base to identify the characters. The following steps are followed for this purpose.

Noise removal: In this process the small unwanted black portions embedded in the white characters are extracted from each sub-image by applying the following algorithm.

ALGORITHM 2
Step 1: Mark each black region by unique level by using region growing algorithm given in ALGORITHM 1.
Step 2: Average number of pixel *A* in each level is determined.
Step 3: If number of pixel in a particular level *L* is less than the average *A*; the pixels with level *L* are converted in to white.

The application of this step is shown in Figure 6.

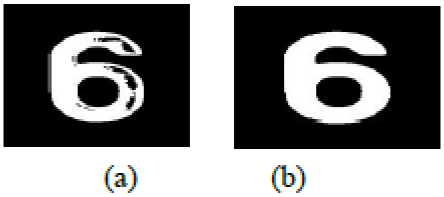

Figure 6: (a) Characters before noise removal. (b) Characters after noise removal.

Thinning: The segmented image is then thinned to get single pixel skeleton of the character. The standard thinning algorithm [10] by using a set of eight structuring element is used for this purpose. Figure 7 shows the thinned character of Figure 6 (b).

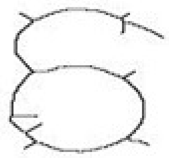

Figure 7: Thinned characters 6(six) of negative of Figure-6

Width Normalization: This step is performed to crop the rectangular area containing the character and eliminating the black space. For these purpose we apply the following algorithm.

ALGORITHM 3
Step 1: Scan the image from left to right to find the column C1 that contains the first black pixel.
Step 2: Scan the image from right to left to find the column C2 that contains the first black pixel.
Step 3: Scan the image from top to bottom to find (a) (b) the row R1 that contains the first black pixel.
Step 4: Scan the image from bottom to top to find the row R2 that contains the first black pixel.
Step 5: Extract the image from pixel (R1, C1) to (R2, C2).

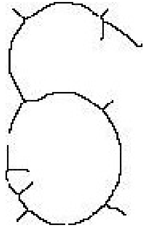

Figure 8: After width normalization of Figure 7

Scaling: After width normalization we get only the rectangular area containing the character. But the size of the character may vary from number plate of different vehicles. We apply standard scaling [5] operation to make all the character of same size.

Character recognition: After Scaling the characters are recognized by the following steps [11].
1. Feature Point Extraction: After preprocessing step, we select a set of representative points, known as feature points, against which the properties of the given character are compared with the features of characters stored in

database. The set of feature points are extracted depending on Centre of Gravity (COG) [10] of the character using the following algorithm.

ALGORITHM 4

```
Step 1: Calculate the COG of the character image.
Step 2: Calculate set of feature points of the image(s) using following steps:
        2.1: Draw a horizontal line through the each COG to break each image into two sub-images.
        2.2: Calculate COG of the signature belonging to each of the sub-image.
        2.3: Draw a vertical line through each of the COG to break each sub-image into two sub parts.
        2.4: Calculate COG of the signature belonging to each of the sub-image of Step 2.3.
Step 3: Repeat Step 2 for N number of times.
Step 4: Consider the COGs obtained in Step 2.2 and Step 2.4, in each iteration, as set of feature points.
```

The above algorithm is illustrated through the following example.

First calculate the COG1 of the signature as in Fig. 9(a). Next draw a horizontal line through COG1 as in Fig 9(b). Find the COG2 and COG3 of the two sub-images as in Fig 9(c). Next draw a vertical line through COG2 and COG 3 as in Fig 9(d). Next calculate COG of each of the four sub-images and draw horizontal line through each of the COG as in Fig 9(e).

Perform these steps repeatedly. Since after each subdivision (horizontal or vertical) number of COG is doubled, we can calculate number of COGs after N subdivision.

At first number of COG is one. After first sub-division we get two new COGs. After second subdivision we get four new COGs and so on. Hence number of COGs after N subdivision is $2^0+ 2^1+ 2^2+ ….. + 2^N= 2^{N+1}-1$.We consider all COGs except the initial COG as feature points. Hence number of feature points is $2^{N+1}– 2$.

Applying ALGORITHM 2 we generate set of feature points and consider this set as horizontal set of feature points H. Similarly we generate another set of feature points and consider this set as vertical set of feature points V just interchanging Step 2.1 and step 2.3 of ALGORITHM 2.

The feature point extracted for the digit 6 is shown in Figure 10.

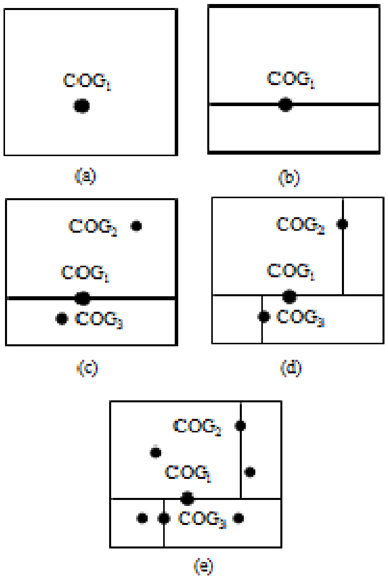

Figure 9: Calculation of feature points

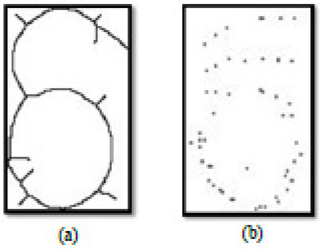

Figure 10: (a) Scaled image (b) 60 feature points

2. Personal Feature Extraction: After finding two sets of feature points, the properties of the character are extracted. These properties are then use to extract the ASCII code of the character. We extract the properties using three following steps.

Construction of the Complete Bipartite Graph: A weighted complete bipartite graph is constructed by the set of points in H and V. If i is point such that i Є H and j is point such that j Є V then there exits an edge E_{ij} with cost W_{ij} such that

$$W_{ij} = D_{ij} + (\text{number of black pixel on eight neighbor of i})^N + (\text{number of black pixel on eight neighbor of j})^N$$

where D_{ij} is the Euclidian distance between points i and j and N be the number of subdivisions used in feature point extraction. Since W_{ij} will be used for matching the given character with the characters stored in the database, small difference of the properties should result large difference in W_{ij}. Hence we use power function in W_{ij}.

Construction of Adjacency matrix: From the weighted graph we form an adjacency matrix. Since we have two set of points H and V with $2^{N+1} - 2$ points in each set we construct an adjacency matrix A of size $(2^{N+1} - 2)$ X $(2^{N+1} - 2)$ such that

A_{ij} =1 iff i Є H and j Є V and Wij is minimum for all j
= 1 to $(2^{N+1} - 2)$
= 0, otherwise.

Feature extraction: Finally, the personal feature of the character is extracted by computing the angle of each edge present in the adjacency matrix with the COG of the signature by the formula

$$A = COS^{-1} ((b^2 + c^2 - a^2) / (2bc))$$

considering that the two end points of the edge form a triangle with COG as shown Fig 3. Since there are $(2^{N+1} - 2)$ number of edges in the adjacency matrix there will be $(2^{N+1} - 2)$ number of different angles.

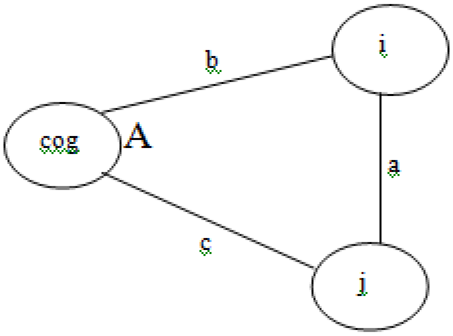

Figure 11: Angle calculation

3. Classification: Before classification step, feature of the character set used in number plate {A, B, …., Z, 1, 2,…, 9} i.e. $(2^{N+1} - 2)$ number of angles are extracted for each character and stored into the database.

After this, a character to be recognized is accepted and $(2^{N+1} - 2)$ number of angles of the given signature is extracted as feature points using same approach. Next the given character is matched with the features of characters stored in the database by the following algorithm.

ALGORITHM 5

```
Step 1: For each character $i$ stored in the database
   Do
     Matched $i = 0$
     For each angle $a$ of the $i$-th character stored in the database
     Do
```

```
        Matched $i$ = Matched $i$ + Difference of $a$ with corresponding angle of the character to be recognized
      End for
    End for
Step 2: Find $k$ such that Matched $k$ is minimum.
Step 3: Recognized character is $k$
```

IV. RESULT

The proposed algorithm is applied on various images of number plates and the recognized numbers from plate images is satisfactory. Table 1 describes the strength of the proposed method.

Table 1: Performance of the proposed algorithm

Type of Image	No. of images	Total No. of characters in the image	No. of Character recognized	Percentage of recognition
Clear image	120	1080	960	88.9
Blurred image	50	450	220	49
Skewed image	30	270	108	40
Average image	30	270	178	66

V. CONCLUSION

The proposed algorithm performs well for clear images, i.e. for the image of the number plate taken for standing vehicles. But if the image is taken for a moving vehicle, the performance degrades. Thus in future, more concentration will be given on work involving moving vehicles.

REFERENCES

[1] Central Motor Vehicle Rules 1989, Rule-50(2)(d). Government of India - Department of Road Transport & Highways. 1989. p. 4. Retrieved 2009-03-15.

[2] Wawage, P., Oza, S.: Design Approach for Vehicle License Plate Automatic Detection and character Recognition System Using Classification Algorithm, International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 10, October 2012.

[3] Khalifa, O., Khan, S., Islam, R., Suleiman, A.: Malaysian Vehicle License Plate Recognition, The International Arab Journal ofInformation Technology, Volume 4, No. 4, October 2007.

[4] Bharath, D.K., Chandrashekar, V.E.: Vehicle Number Segmentation and Recognition by Using Region Properties and Thresholding, International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064 pp. 308-311.

[5] Roy, A., Ghoshal, D.P.: Number Plate Recognition for Use in Different Countries Using an Improved Segmentation, IEEE.

[6] Parking and Traffic Technologies Ltd, Smartreg ANPR, Available: http://www.parkingandtraffic.co.uk/ANPR/smartreganpr

[7] Nijhuis, J.A.G., Brugge, M.H., Helmolt, K.A.: Car License Plate recognition with network and fuzzy logic, in Proc. Of IEEE International Conference on Neural Networks., volume 5, pp 2232-2236, Dec 1995

[8] Chang, L., Chen, L., Chung, Y., Chen, S.: Automatic license plate recognition IEEE Transaction Intelligent Transportation System, 5:42-53,2004

[9] Otsu, N.: A threshold selection method from gray-level histograms, IEEE Trans. Sys., Man., pp.62–66.

[10] Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd Edition, Prentice Hall, 2009.

[11] Ghosh, B.R., Banerjee, S., Dey, S., Ganguli, S., Sarkar, S.: Off-Line Signature Verification System Using Weighted Complete Bipartite Graph, 2nd International Conference on Business and Information Management (ICBIM), ISBN : 478-1-4799-3264-1/14/$31.00 ©2014 IEEE pp.109-113.