IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-12, Dec 2018 E-ISSN: 2347-2693

# Automated Number Plate Recognition Using Template Matching

**Nancy Aggarwal[1*], Shilpa Sethi[2]**

[1]Department of Computer Engineering, YMCA University of Science & Technology, Faridabad
[2]Department of Computer Application, YMCA University of Science & Technology, Faridabad

*Corresponding Author: nantayal94@gmail.com

**Available online at: www.ijcseonline.org**

Accepted:15/Dec/2018, Published: 31/Dec/2018

***Abstract***- Automated number plate recognition is an image processing technique that uses number plate for identification and authorization of products. The paper aims to design a vehicle identification system by using template matching. The proposed system detects an authorized vehicle followed by capturing its image. The system is highly efficient and can be installed in high security zone like government offices including parliament and Supreme Court. The system is implemented in MAT lab. The experimental result of proposed system is also compared with existing pattern matching technique.

***Keyword*** – Automated Number Plate Recognition (ANPR), Template Matching.

## I. INTRODUCTION

The automated number plate recognition (ANPR) is a digital image processing technique. This technique is mainly focus on image and it developed computer systems which perform the operation on an image. This system take a digital image as an input and the computer system process that image with some algorithm, and gives some information as an output with the help of OCR technology, image processing technology is used in number plate recognition extraction which gives the digital image of the vehicle .

The number plate recognition plays important role in extraction, processing, analysis and recognition. The utility of this system includes toll tax, parking area, security area, and boarder area. It also assists in maintaining traffic rule and legal law. The extraction

of number plate automatically captures the image from the moving vehicle. He paper is organised as follows: The Section I contains the introduction of Automated Number Plate Recognition using Template Matching, Section II contains the related work of ANPR using Template Matching, Section III contains the purposed architecture and essential steps of ANPR using Template Matching, Section IV explain the performance analysis of ANPR using Template Matching. Section V contains conclusion and future scope of ANPR using Template Matching.

## II. RELATED WORK

Digital Image processing is widely used technology, which convert scanned image into a useful information. The digital image processing is take image as the processing. The printed text using segmentation and then convert it into machine encoded text such as ASCII representation.

i. It is proposed a system that works on Korean plate. This system implementing for impressive average character recognition and Support Vector Machine [1]. It designed most optical character recognizer worked in a 2-D plane are capable to manage high successful rate only for shooting distance and visual angle within an average range [2] [3].

ii. It designed a two hybrid recognition system that combination of statistical and structural method. This technique Distinguished similar characters using local structural features. The first step, independently recognize the character by the four sub-classifier and then intermediate result were combined using Bayes theorem. The second step, remove the ambiguity, if exist among the character [4].

iii. It designed a system that extracts the number plate using video camera. . The recognition of number plate is done by template matching. To locate the probable number plate using gradient operator, the image is binarize using otsu method [5]. The binary image and prototype similarity is calculated using the root-mean-squared-error (RMSE)[7].

iv. It designed a novel algorithm that identifies vehicle chassis number identification system. This system is done using artificial neural network. This method provide correct identification rate with considerably high value [6].

Based on the available literature the proposed methodology is discussed in next section.

## III. AUTOMATED NUMBER PLATE RECOGNITION USING TEMPLATE MATCHING

The digital image processing in ANPR provides the recognition of optical character. The architecture of the Automated Number Plate Recognition (ANPR) is shown in figure 1. It consists six modules which are image capturing module, gray scale conversion module, bitmap convertor, word segmentation module, template loader, template processor.

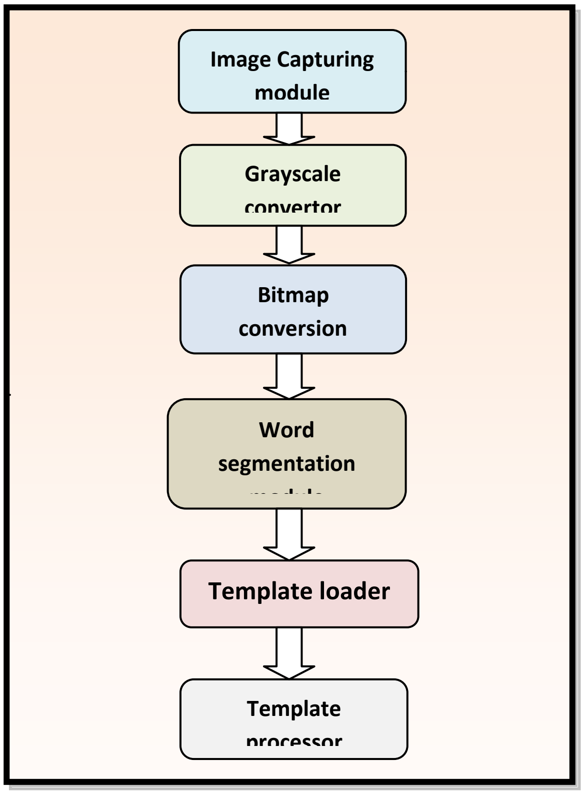



**Figure 1: Proposed Architecture of ANPR using Template Matching**

The proposed system work as follows: First of all, a database of templates is prepared. Each Template is prepared using Microsoft paint in proposed system. Image is capture using webcam. Then Input image is transform into a grayscale image (black & white image). The grayscale image then converted into a bitmap image that stores each pixel with two values 0 or 1. The characters of the image are separated into individual character**.** The individual character is match with every template image for best matching. Recognize the optical character from the number plate and print the output as a string of character. Detail working of each of these module is given in following subsections.

### a. Image Capturing Module

Image capturing is most basic and indispensible step in image processing. The choice of capturing image through webcam is due to its facility to produce easily machine editable image. The image of vehicle number plate is capture through the webcam. The captured image can be stored in specific format like jpeg, bmp, tif, png etc. The algorithm of image capturing module is outlined in Figure 2.

**Image_capturing()**

**INPUT:-** capture the vehicle number plate.

**OUTPUT:-** Extract the number plate.

**METHOD:-** 1. if(webcam()== true) {

1.1 shoot a picture using webcam;

1.2 read (shooted picture); }

2. else { 2.1 Delete the camera object;

2.2 Load the picture from the dataset;

2.3 Select picture format;

2.4 Read (selected picture) }

**Figure 2 Algorithm of Image Capturing**

### 3.2 Grayscale Conversion Modules

The inherent complexity is more storage requirement of color impetus to convert them into gray image. The RGB2GRAY scale function is used for convert RGB image into grayscale.

In the grayscale image the each pixel value is a single sample per pixel. The single sample per pixel is representing only an amount of light. The grayscale image is also known as gray monochrome or black & white. Algorithm of Grayscale Image converter is given in Figure 3.

**Grayscale_convertor()**

**INPUT:-** Extracted image represented as '**f**' from image capturing module.

**OUTPUT**:- gray scale image represent by '**g**'.

**METHOD**:- 1. Read **f**;

2. Resize the image aspect ratio=400,NaN;

3. Call rgb2gray();

4. Display **g**.

**Figure 3 Algorithm of Grayscale Conversion**

### 3.3 Bitmap Conversion Module

The purpose of bitmap conversion is twofold: first facilitate the segment the gray scale image into individual word by converting 24 different gray shade into single bit representation i.e either 0 or 1. Second, the OCR package available in market also works on bit-level. Algorithm of Bitmap Conversion module is shown in Figure 4.

### 3.4 Word Segmentation Module

In this process, every individual character is separated. This isolates the different sub-parts of an image. It separates pixels of an image as per the content in data like words, paragraph etc. Segmentation of fixed-pitch fonts accomplished relatively simply by align the image to a uniform grid

**Bitmap_conversion()**

**INPUT:-** gray scale image **g**;

**OUTPUT:-** bitmap image represent as '**b'**;

**METHOD:-** 1. Read **g**;

2. G=double(g);

3. Sum of all gray level's pixel of all grayscale image

**For** i←1 : x

**For** j←1 : y

Sum ← sum + g( I , j );

5). Calculate Threshold value← insert mathematical object;

6). Image array having same number of row and column with all element as 0;

7). Iterate over all the pixels of grayscale image and assign 1 to to binary (i , j);

8). If (g(i , j)>=(threshold);

value [i , j] ← 1;

8.1). else value [i , j] ← 0;

**Figure 4 Algorithm of Bitmap Conversion**

based on where vertical grid lines will least often intersect black areas. From the entire image, Work segmentation retrieves the information. This information may be a line, character and even a word. The word segmentation is separate every individual character and the separated character is use for read the character from the image. After read each character the template matching is used for best matching.

### 3.5 Template Loader

The templates are create with the help of Microsoft paint application. After the word segmentation, each word is matching with template. This process involves the use of template. Database of characters or templates are load all the possible input character that are useful for template matching.

The current input character is compare to each template to find either an exact match or the template with the closest representation of the input character.

The template is load 0-9 and A-Z input character and each input character loads with unique index value. When current input character is compare with each template, this process perform using index value of each template. All templates with its name is shown in Figure 6.. Algorithm of Loading Template in database is outlined in Figure 5.

**Template_loader()**

**INPUT:-** 0-9 or A-Z.

**OUTPUT:-** templates.

**METHOD:-** 1. Load (A to Z) ← index(1 to 33);

2. load (1 to 9)← index(34 to 46);

3. if (value = 1 to 46) {

Print template character; }

else {

Print 0; }

**Figure 5 Algorithm of Template Loader**

The Figure 6 is show 0 to 9 and A to Z templates and each template is stored with unique indexing value. These templates are used for producing the output. The index value is match with input character until find the best matching template.

### 3.6 Template Processor

Each pixel has two coordinate x and y. individual character act as an input character (i). The number of template is denoted by "Tn" and template character is also have a two coordinate x and y. String is denoting by S. Template processor is use for template matching with input character. If $I_m(X_m , Y_m)$ is the input character and $T_n(X_n,Y_n)$ is the template n, then the matching function $S(I, T_n)$ will return a value

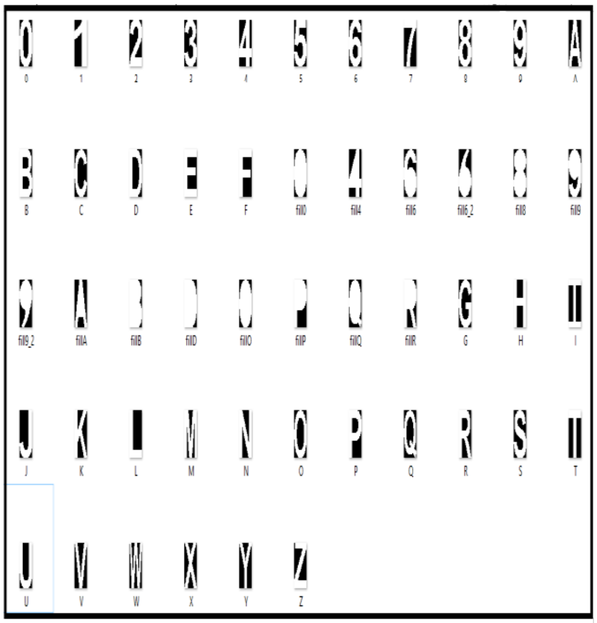


**Figure 6 Templates**

indicating how well template n matches the input character. Algorithm of Template Processor is outlined in Figure 7.

**Template_processor()**

**INPUT:-** individual character of number plate(c), Templates (t)

**OUTPUT:-** best matching result(w).

**METHOD:-** If (t ← c)
Print the number plate variables in notepad;
else
Print unable to extract the variables of number plate;

**Figure 7 Algorithm of Template Processor**

The number plate recognition is first extract the RGB image using webcam/video camera. The RGB image of number plate is recognized then the RGB image Is convert into grayscale image with the help of rgb2gray function. The grayscale image has two color that is black and white and a single sample per pixel. The grayscale image is converting into a bitmap image. the bitmap image which has two value(0 or 1) for each pixel. The bitmap image is use for word segmentation. The word segmentation part is separated each word from the image. the each word is compare with template matching. Template processor gives the string of character for number plate identification.

## IV. PERFORMANCE ANALYSIS

The printed text from vehicle number plate using template matching is implemented in MATLAB. The use of MATLAB the image of vehicle number plate is extract using webcam. The extracted image is convert into grayscale using **rgb2gray** MATLAB function. The grayscale image has two color that is black and white and a single sample per pixel. The grayscale image is converting into a bitmap image. The bitmap image which has two value (0 or 1) for each pixel.
The grayscale image is converted into bitmap image using MATLAB functions. The bitmap image is separate all optical characters. The segment character is match with template then appropriate match gives the output as a string of character in the Microsoft Notepad. The matching results are summarized in Table 1.

**Table 1 Matching Result**

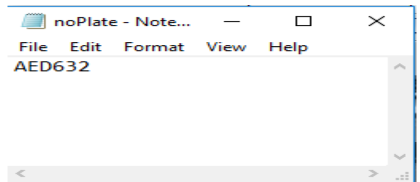
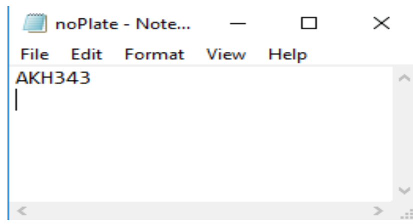
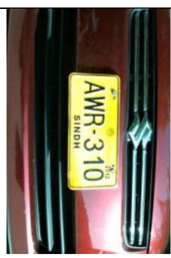
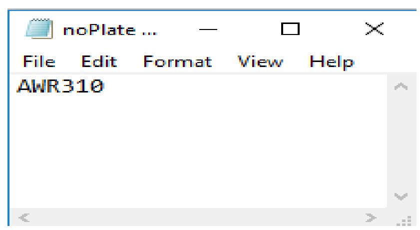
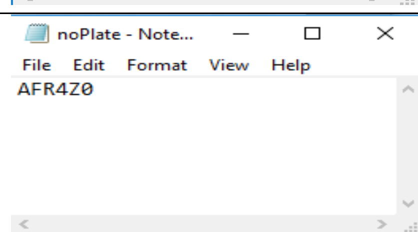
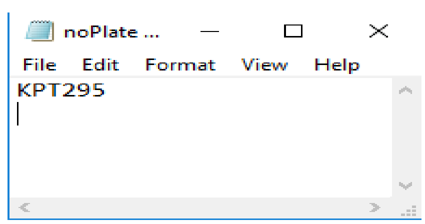
| S. no | Image of vehicle number plate | Actual number | Result | Remark |
|---|---|---|---|---|
| 1. |  | AED632 |  | **Correct** |
| 2. |  | AKH343 |  | **Correct** |
| 3. |  | AWR310 |  | **Correct** |
| 4. |  | AFR420 |  | **AFR4*0 (One character is not correct)** |
| 5. |  | KPT295 |  | **Correct** |

Table 1 contains five images and these five image contain 14 letters (0, 1, 2, 3, 4, 5, 6, 9, AED, AKH, AWR, AFR, KPT). Each letter contains number of test samples. 20 samples of 0, 16 samples of 1, 24 samples of 2, 22 samples of 3, 67 samples of 4, 40 samples of 5, 59 samples of 6, 45 samples of 9, 12 samples of AKD, 15 samples of AKH, 8 samples of AWR, 5 samples of AFR, 5 samples of KPT. This template matching technique is compare with pattern matching technique. The arrangement of entities and objects is pattern. The table 2 is show that matching template result of both techniques:

**Table 2 Matching Output**

| **Letter Capture** | **Test Sample** | **Pattern Matching Sample recognize result** | **Template Matching Sample recognize result** |
|---|---|---|---|
| **0** | 20 | 16 | 19 |
| 1 | 16 | 12 | 14 |
| 2 | 24 | 20 | 21 |
| 3 | 22 | 15 | 18 |
| 4 | 67 | 50 | 62 |
| 5 | 40 | 34 | 38 |
| 6 | 59 | 45 | 51 |
| 9 | 45 | 30 | 35 |
| AED | 12 | 8 | 11 |
| AKH | 15 | 12 | 14 |
| AWR | 8 | 6 | 7 |
| AFR | 5 | 2 | 4 |
| KPT | 5 | 3 | 4 |

The table 2 shows that letter have number of test samples. This table 1 provide the recognize result of all letters using both technique. After recognition results, the accuracy in percentage is define in table 2 the accuracy is calculate by recognition sample of each letter divide by total sample value and multiply by 100.

$$Accuracy\ in\ (\%) = \left(\frac{Recognition\ Sample}{Total\ Sample}\right) * 100$$

This formula to calculate the accuracy is done for each letter with both techniques. The table 3 shows the accuracy and the accuracy is in percentage(%). The Pattern matching accuracy result in between 40% to 87%. The Template Matching accuracy result in between 77% to 95%. It is observed that template matching accuracy is better than pattern matching. Table 3 shows the relative accuracy of both techniques.

**Table 3 Performance Analysis Results**

| **Letter Capture** | **Pattern Matching Accuracy (%)** | **Template Matching Accuracy (%)** |
|---|---|---|
| 0 | 80% | 95% |
| 1 | 87.5% | 90.67% |
| 2 | 83.33% | 87.5% |
| 3 | 68% | 85.7% |
| 4 | 74.6% | 92.5% |
| 5 | 85% | 95% |
| 6 | 76% | 86% |
| 9 | 66.7% | 77% |
| AKD | 66.7% | 91% |

| | | |
|---|---|---|
| AKH | 80% | 93.33% |
| AWR | 75% | 87.5% |
| AFR | 40% | 80% |
| KPT | 60% | 80% |
| **Average Accuracy(%)** | **72.5%** | **87.7%** |

It may be noted from table 2 that average accuracy of number plate recognition using template matching is calculate more than existing techniques.

## V. CONCLUSION

The purpose of extraction of vehicle number plate is to identify vehicle with its unique number. The proposed technique successfully ideates the vehicle with average accuracy of 87%. In future, number plate identification can be done faster by storing string value of plates. It will result not only in optimizing the speed but also result in less storage requirement.

## REFERENCES

[1] M.Bhargavi, Sajja.Radharani, "*Car License Plate Detection Using Veda*", International Journal of Scientific Research in Computer Science and Engineering, Vol.5, Issue.6, pp.19-26, 2017.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Swati Sinha, Rakesh Bharati, "*An Efficient and Practical Approach to License Plate Localization and Binarization*", International Journal of Science.

[7] Bhoomika and S. Sethi "Analysis of various Semantic analysis techniques" " International Journal of Creative and Research thoughts. Vol- 6 • 2018 •ISSN 2320-2882

### Authors Profile

Ms. Nancy Aggarwal pursed Bachelor of Computer Application from Saraswati Mahilla Mahavidyalaya Palwal affiliated from Maharshi Dayanand University, Rohtak in 2015 and Master of Computer Application from MVN University of Palwal, in 2017. She is currently pursuing Master of Technology (Information Technology) from YMCA University of Science & Technology of Faridabad in Department of Computer Engineering.

Dr. Shilpa Sethi has completed her Ph.D in 2018 from YMCA University of Science and Technology, Faridabad. She has received her Master in Computer Application from Kurukshetra University, Kurukshetra in the year 2005 and M. Tech. in Computer Engineering from MD University Rohtak in the year 2009. She has published more than 20 research papers in various International journals and conferences. Her area of research includes Internet Technologies, Web Mining and Information Retrieval System.