

# **An Improved Template Matching Algorithm for Car License Plate Recognition**

**Shreeja Chakraborty**  
School of Education Technology  
Jadavpur University  
Kolkata

**Ranjan Parekh**  
School of Education Technology  
Jadavpur University  
Kolkata

## **ABSTRACT**

An improved Template Matching algorithm is proposed to recognize license plate characters. The license plate recognition system has two main parts: Segmentation of license plate characters and Recognition of characters from the plate. Character Segmentation is performed using connected component analysis. For recognition of number plate, a new algorithm is proposed, which uses dynamically generated license plate characters as database template and recognition is done using correlation technique. The algorithm was tested with 100 images of Indian Vehicles taken under various background conditions. The experimental result shows that accuracy obtained is comparable to the best figures reported in literature.

## **General Terms**

Image Processing, Pattern Recognition, Automatic Number Plate Recognition, Template Matching

## **Keywords**

Correlation, Template Matching, Character Segmentation, License Plate Recognition

## **1. INTRODUCTION**

Automatic License Plate Recognition is an important technique used in Intelligent Transportation System, used to identify vehicles by their numbers without direct human interventions. It is mainly used for monitoring, surveillance and security and also for law enforcement. Several applications of Automatic Number Plate Recognition, makes it an important area of research.

Currently, vehicles are used widely in all parts of the world. With the rapid increase in the number of vehicles, traffic violations are more likely to occur. In order to keep track of the vehicles, license plate can be used, which provide unique information for every car. Thus, automatic collection and management of license plate information can be used effectively to supervise public transportation.

However, the task of recognizing Indian vehicle is bit difficult, since the license plates are not standardized across different states. Indian license plate can be single row or double row. It can also be written in multiple scripts.

The main purpose of the paper is to design and create a robust system to recognize license plate characters from a car image. Work focuses on designing a system that uses dynamically generated database templates to overcome the shortcomings of template matching recognition technique which includes sensitivity to noise and style variations and inability to handle rotated characters.

## **2. RELATED WORKS**

Several methodologies have been proposed for license plate recognition system. Some of them have a good accuracy but have more complexity than the others. A section of the works that focuses on Template Matching as recognition technique are discussed below:

[1] proposed a new algorithm for recognition of Indian license plate using morphological operation, thresholding operation, edge detection, bounding box analysis for number plate extraction, character separation using segmentation and character recognition using template matching method. [2] deals with methods achieving invariance of systems towards image skew using homography and various plate conditions. Recognition is done using template matching with correlation technique. The comparison process is simplified by semantic analysis of number plate patterns. [3],[4],[10] presents a real time and robust method of license plate recognition from cluttered images based on mathematical morphology and template matching. In this system, the main stage is isolation of license plate from car image taken by digital camera under several circumstances like illumination, slope, distance, angle etc. In [5], several traditional image processing techniques such as Image Enhancement, unsharp masking, edge detection, filtering and component analysis are used for plate localization. Character segmentation is done using connected component analysis and recognition using template matching. By setting template priority, execution time is reduced. When higher priority template is matched, lower priority template is ignored. In [6], morphological operation is applied on image to locate plate. Then plate region is preprocessed using histogram equalization technique. The smearing and morphological algorithm is used to segment characters and recognition is done using template matching. [7] identifies factors that affects effectiveness of template matching in recognizing license plate characters. These factors are font types, template resolution, noise, tilting etc. The proposed system improves performance by using more relevant font types, construction of new template with higher resolution and removing more noise. [8] is designed for grayscale images so it detects number plate regardless of its color. Template matching is done for character recognition and it is implemented for automatic toll tax collection. [9], focuses on solution for image disturbance resulting for uneven illumination and various outdoor conditions such as shadow and exposure. Improved Bernsen algorithm detects license plate from car image and recognition is done using direct template matching.

## **3. PROPOSED APPROACH**

The block diagram of the proposed system is given:

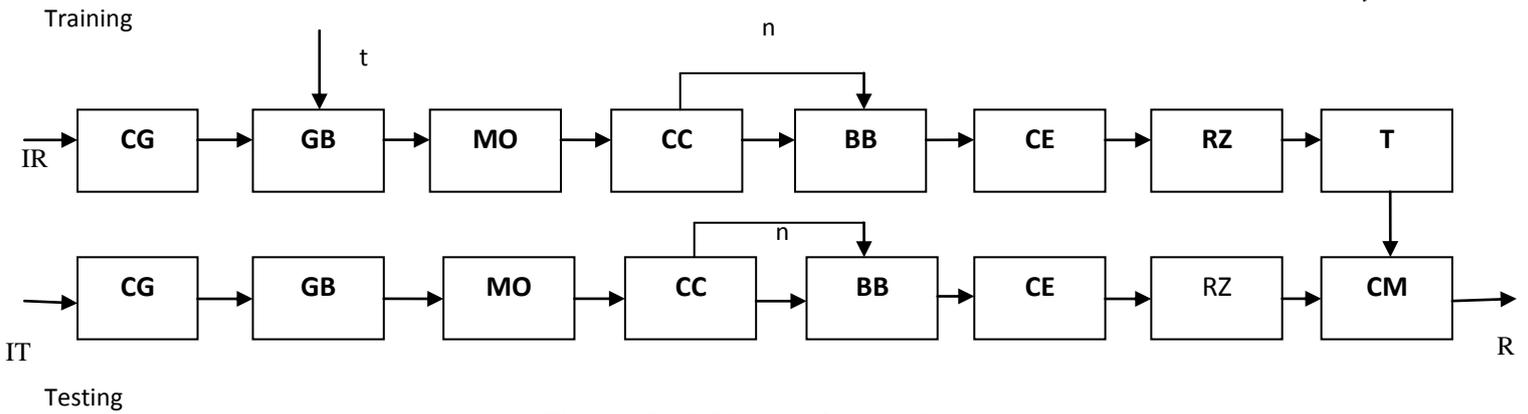


Figure 1: Block Diagram of proposed system

Where IR : input training image, IT : input testing image, CG : color to grayscale conversion, GB : grayscale to color conversion using threshold  $t$ , MO : morphological open operation to remove noise, CC : connected component labeling to identify no. of characters  $n$ , BB : generation of boundary boxes around each character, CE : extraction of individual characters based on boundary box coordinates, RZ : resizing of extracted characters to standard dimensions, T : insertion into template, CM : comparison of training and test characters using 2D correlation, R : Output characters printed as result.

The proposed system can be divided into following phases: Pre-Processing Phase, Training phase, Testing Phase and Classification Phase. These phases are described in greater details in following subsections.

### 3.1 Pre-Processing Phase

In this proposed approach, the image is at first captured using camera in ideal lighting conditions with sufficient resolution. Input images are manually cropped to isolate the number plate discarding the surrounding non-relevant areas.



Figure 2: Car image



Figure 3: Cropped license plate from the car image

The cropped license plate is preprocessed to enhance the quality of the image. It is an important step required to facilitate recognition process.

The first step is RGB to grayscale conversion, which forms the foundation of latter steps. The conversion is performed using following formula:

$$A_{GL} = (3A_R + 6A_G + A_B)/10 \quad (1)$$

Where  $A_{GL}$  is the converted grayscale image and  $A_R$ ,  $A_G$  and  $A_B$  are R, G, and B spectrum of color image respectively.

Then the grayscale image is converted to binary using appropriate threshold  $T$ , to highlight the edges in black background using equation (2):

$$G(x, y) = \begin{cases} 1, & \text{if } A(x, y) \geq T \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

### 3.2 Training Phase

The binary image of a number plate essentially contains the alpha-numeric characters but can also contain other non-desirable portions considered as noises and need to be removed e.g. screws on the plate. To remove these relatively small image objects, the binary image is subjected to morphological 'open' operation by specifying an experimentally determined value  $v$  such that all objects containing number of pixels smaller than specified value is removed.



Figure 4: Binary image of car plate



Figure 5: Binary image after open operation

To determine the total number of alpha-numeric characters in the number plate, a connected component labeling using an 8-level connectivity is performed. This returns the total number of characters  $n$  in the plate. Based on the connected components the bounding box around each character is also determined. Total number of bounding box is more than the number of characters. In order to determine number of characters, we are ignoring the first bounding box indicating boundary of license plate. The bounding box is used to extract each single character from the plate which is then resized to standard dimensions of  $a \times b$  for subsequent insertion into training template.

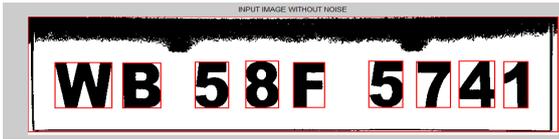


Figure 6: Characters are detected from plate

The current work has been designed specifically for Indian number plates. The number of alpha-numeric characters in an Indian number plate can either be 8, 9 or 10. Accordingly three different templates are designed, the first for 8, the second for 9 and the third for 10 characters. The first template T1 is therefore a 2D matrix with 8 columns and the number of rows corresponding to the total number of plates containing 8 characters, say N1. The second template T2 likewise contains 9 columns and N2 rows for the number of plates containing 9 characters, and the third template T3 containing 10 columns and N3 rows for the number of plates containing 10 characters. The strings of license plate characters are stored in 10 different variables Plate1, Plate 2, Plate 3 and so on which will be required for classification phase.

### 3.3 Testing Phase

The binary image of a test number plate is subjected to the same pre-processing steps and the connected component labeling as the training set images. Each character from the test plate is extracted and resized to pre-determined standard dimensions. Each character is then matched with the training templates using a classification procedure.

### 3.4 Classification

The number of characters in a test plate is extracted during the testing phase. Based on this one of three training templates is chosen for classification. The test character is compared with each character in the chosen template using the process of 2D correlation. The correlation coefficient is computed using following formula:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (3)$$

Where A and B are images to be compared,  $\bar{A}$  computes mean of the values in A and  $\bar{B}$  computes mean of the values in B and subscript indices m and n refers to the total number of pixels in the image.

Within the template each row represents a training plate. As the test character is matched with each character of each row of the template, a correlation score is generated for each match and summed up for each row. At the conclusion of the matching process, a match vector is generated with its number of elements equal to the number of rows in the template and the value of each element represents the matching score with

each row. Ideally only one element should be non-zero which indicates the matched training class and the corresponding index of the element is returned by the system. In case more than one element of the matching vector is non-zero it represents that different characters of the test plate has matched with different training plates. In such a situation the position of the maximum value in the matching vector is considered to indicate the training plate which provides the maximum matching score for the test plate. And finally, the license plate is returned (which is saved in a variable), whose corresponding matching score is maximum, and class to which the plate belongs is assigned.

A threshold is defined to determine whether recognition of plate is correct or not. If the value of matching score is less than threshold value then the plate image is sent to digit recognition module. For digit recognition, we create a 10 number database template and last 4 digits of unknown plate image are compared with database template and correct plate is identified.

To calculate the accuracy, a counter variable is defined and is initialized to zero. When a number plate is successfully recognized, counter value is incremented by 1. Recognition accuracy is calculated using following formula:

$$\text{Recognition Accuracy}(\%) = 100 \times \left(\frac{C}{T}\right) \quad (4)$$

Where C is the total number of successful recognition of license plates and T is the total number of license plates used for testing.

The output is printed to text file as follows:

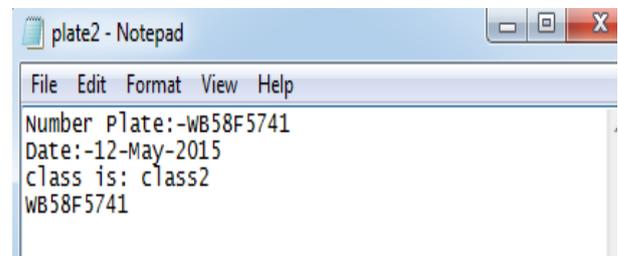


Figure 7: Format in which results are displayed

## 4. EXPERIMENTAL RESULTS

For performing the experiment, 10 classes of license plate image are captured using 5 mega pixel mobile camera and are saved in ".jpg" extension. Each class consists of 10 slight variation of single car image. Thus, a dataset containing 100 images of car plate is used for conducting the experiment. 1 image per class i.e. 10 images of license plate is used for creating the database template. 9 images per class i.e. 90 images of license plate are used for testing. The full experimentation is performed in Matlab.

### 4.1. Training Phase

1 image per class is used for training purposes. At first license plate images are converted to binary, characters are segmented and resized. And segmented character images are store in 2d array in form of database template. Images of some license plates that are used to create database template and images of segmented database characters are shown below



Figure 8: Images used for Training

The images of three database template are shown below:

Template 1

**BR1Z1971**

Template 2

**W B 5 8 F 5 7 4 1**  
**W B 1 2 C 6 6 9 9**  
**W B 0 6 H 0 8 7 5**  
**W B 0 6 H 6 5 6 7**  
**W B 1 8 C 2 4 0 0**

Template 3

**W B 0 2 A F 6 1 4 6**  
**W B 0 2 A A 6 3 0 4**  
**W B 0 2 A D 7 6 5 8**  
**W B 0 2 A C 8 7 4 9**

Figure 9: Database Template

#### 4.2. Testing Phase

9 images per class are used for testing purposes i.e. 90 images of license plate are tested to conduct the experiment. The images are preprocessed; characters are segmented and sent to recognition module for classification.

Three out of nine images that are used for testing are shown below:



Figure 10: Images used for Testing

#### 4.3. Classification

The test images are compared with the training images using template matching to assign the class and compute the accuracy. 90 out of 90 plates are successfully recognized using our proposed algorithm. The recognition rate achieved using proposed template matching method is 100%. The time required to test 90 images is 2 minute 58 seconds and time required to generate the templates is 17 seconds. Thus, total execution time is 3 minutes 15 seconds. Based on these results, it is shown that the proposed algorithm is very effective for recognition of number plates.

Some sample calculations for recognition of license plate is shown:

For training, 3 templates are designed; T1 containing 8 digits license plates, T2 containing 9 digits license plates and T3 containing 10 digits license plates. Each class contains 9 testing samples.

##### Case 1: One element of matching vector is non-zero (ideal case)

Consider first testing sample of class 5 is used for sample calculations. The number of characters ( $n$ ) is extracted during the testing phase. Here,  $n = 9$ , so template T2 is loaded. T2 contains 5 license plates of 9 digits characters.

Now each character from test plate are extracted and resized. Consider First segmented character from the test plate. The correlation coefficient is computed between this unknown character and database characters and the correlation values are stored in an array. Some of the correlation values are shown in the table along with their index positions ( $vd$ ).

**Table 1: Correlation values between test image and database image**

<i>vd</i>	9	10	11	12	13
<b>Correlation values</b>	0.0623	1.0000	0.2392	0.0096	0.0238

For  $vd = 10$ , correlation value is maximum. Since value of  $vd$  lies between 10 and 18, thus counter corresponding to plate2 stored in T2 is incremented.

The same procedure is repeated for the rest of the segmented characters and the counter values along with their corresponding license plate are given in tabular form:

**Table 2: Counter values and their corresponding plates**

<b>Plates</b>	Plate 1	Plate 2	Plate 3	Plate 4	Plate5
<b>Counter values</b>	0	9	0	0	0

The counter value corresponding to the Plate 2 has maximum value, so, string of characters stored in Plate 2 is returned and class to which Plate 2 belongs is assigned.

**Case 2: More than one element of matching vector is non zero**

Consider first testing sample of class 3 is used for sample calculations. The number of characters ( $n$ ) is extracted during the testing phase. Here,  $n=10$ , so template T3 is loaded. T3 contains 4 license plates of 10 digits characters.

Now each character from test plate are extracted and resized. Consider First segmented character from the test plate. The correlation coefficient is computed between this unknown character and database characters and the correlation values are stored in an array. Some of the correlation values are shown in the table along with their index positions ( $vd$ ).

**Table 3: Correlation values between test image and database image**

<i>vd</i>	1	2	3	4
<b>Correlation values</b>	0.8611	0.2210	0.0761	0.0737

For  $vd = 1$ , correlation value is maximum. Since value of  $vd$  lies between 1 and 10, thus counter corresponding to plate1 stored in T3 is incremented.

The same procedure is repeated for the rest of the segmented characters and the counter values along with their corresponding license plate are given in tabular form:

**Table 4: Counter values and their corresponding plates**

<b>Plates</b>	Plate 1	Plate 2	Plate 3	Plate 4
<b>Counter values</b>	8	0	0	2

The counter value corresponding to the Plate 1 has maximum value, so, string of characters stored in Plate1 is returned and class to which Plate 1 belongs is assigned.

**5. DISCUSSIONS**

There are several factors that affect the effectiveness of template matching algorithm like font types, template resolution, noise, tilting etc. So instead of using fixed template (containing 26 letters and 10 numbers) as in [1] [6] [8], the license plate characters are segmented and stored in a cell array and saved as template. When we use 36 character database template for recognition of license plate (using our dataset), 21 out of 90 plates are successfully recognized and accuracy is as low as 23.33%.

In place of using single character template consisting of 93 characters, 3 separate database template are created and loaded for 8 digits, 9 digits and 10 digits plates respectively. This reduces execution time, since unknown segmented image has to be compared against 8 characters (for 8 digit plate), 45 characters (for 9 digit plate) and 40 characters (for 10 digit plates) instead of 93 characters. Execution time reduced from 2 minute 20 seconds to 1 minutes 53 seconds and accuracy increased from 77% to 90%. To further improve the result, threshold is applied and digit recognition is performed and accuracy increased from 90% to 100%.

The system can detects whether a plate is known or unknown based on predefined threshold value. A threshold value determines whether a plate can be recognized or not.

In [1],[3],[4],[10] segmented characters are compared against database characters and correlation values are stored in an array, then letter linked to the index position of maximum value in the array are returned and stored in a output variable. This procedure is repeated for all unknown segmented characters and ultimately the output variable is written to a text file. The shortcoming of this approach is if the system fails to identify a single character, the plate cannot be assigned to correct class, and hence accuracy is reduced. To overcome this problem our algorithm is proposed to recognize license plates.

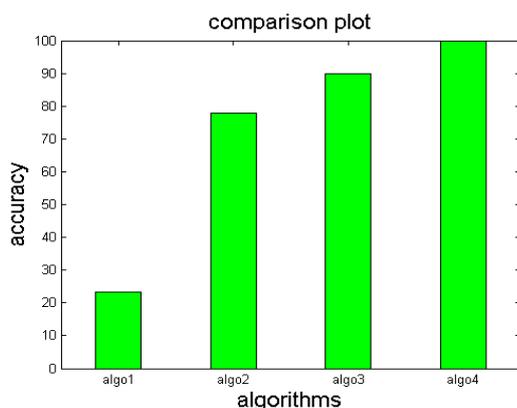
Since digit recognition is performed for validation of license plate recognition, confusions between the conflicting pair of characters like D and 0, B and 8, S and 5 etc are removed. Thus the overall result is improved to a great extent.

The following table gives recognition rate, percentage of accuracy, execution time of various algorithms:

**Table 5: Analysis of different algorithms**

Algorithm No.	Algorithms	Recognition Rate	% of accuracy	Execution time	Discussions
1	Algorithm stated in [1],[6],[8] applied on our dataset	21/90	23.33%	3 minutes 25 seconds	Very low accuracy
2	Single template consisting of 93 characters is used along with our proposed algorithm	70/90	77.77%	2 minutes 20 seconds	Failure of license plate recognition is mainly due to assignment of incorrect plate by the system because all 8 digit plates can have minimum 4 digits common, 9 digit plates can have minimum 5 digits common and so on
3	3 separate database template is used with our proposed algorithm before applying threshold and digit recognition	81/90	90%	1 minutes 53 seconds	Accuracy improved since 3 separate functions are called for recognition of 3 different types of plates
4	3 separate database template is used with our proposed algorithm after applying threshold and digit recognition	90/90	100%	3 minutes 15 seconds	Accuracy further improved by applying threshold and digit recognition

The following bar graph shows comparison of various algorithms in terms of accuracy. Algorithm 1 uses 36 fixed character template for recognition of license plate. This algorithm results in very low accuracy. Algorithm 2 uses single template consisting of 93 characters along with our proposed algorithm for recognition. In this approach, substitution error is common. Algorithm 3 is our proposed method before threshold is applied and digit recognition is performed and Algorithm 4 is also the proposed method after application of threshold and digit recognition.



**Figure 11: Bar graph showing comparison among several algorithms**

## 6. CONCLUSIONS AND FUTURE SCOPE

Thus, an algorithm for segmenting and recognizing the characters from Indian license plate is proposed. At first, the plate area is cropped from the image, then segmentation of characters is done using connected component analysis and finally an improved template matching algorithm is used for recognition of number plate. The improved algorithm uses dynamically generated license plate characters as database template to overcome various problems affecting effectiveness of existing template matching method. The system is tested on 90 images of car, captured using 5 mega pixel camera and taken under various background conditions. The accuracy of current proposed approach is better than those reported in contemporary works. The experimental result shows that the proposed approach produces accuracy of 100% on our dataset. Compared to other methods, this algorithm is fast in execution, efficient in recognition and easy in implementation.

There are many areas which cannot be addressed due to time limitations. For instance, the plate localization module is left out. So in future, this module can be undertaken as an enhancement to the project. Works can also be done on correction of images containing motion blurring and blurring due to the use of low resolution camera. In this proposed approach, template matching is used for character recognition. The character recognition can also be done using feature

based techniques. This can be also a possible enhancement for the project. Also the system cannot handle two row plates that can be addressed in future.

## **7. REFERENCES**

- [1] Ronak P Patel, Narendra M Patel, Keyur Brahmhatt “Automatic License Plate Recognition”, *International Journal of Computer Science and Mobile Computing*, vol. 2, No. 4, pp 285-294, 2013.
- [2] Shishir Kumar, Shashank Agaewal, Kumar Saurabh “License Plate Recognition System for Indian Vehicles”, *International Journal of Information Technology and Knowledge Management*, Vol. 1, No. 2, pp 311-325, 2008.
- [3] Seyed Hamidreza Mohades kasaei, Seyed Mohammadreza Mohades Kasaei “Extraction and Recognition of the vehicle license plate for passing under outside environment”, in proc. of European Intelligence and Security Informatics Conference, pp 234-237, 2011.
- [4] Rajesh Kannan Megalingam, Prasanth Krishna, Prathiesh Somarajan, Vishnu A Pillai, Reswan Ul Hakkim “Extraction of License Plate Region in Automatic License Plate Recognition”, in proc. of International Conference on Mechanical and Electrical Technology, pp 496-501, 2010.
- [5] Sahil Shaikh, Bornika Lahiri, Gopi Bhatt, Nirav Raja “A novel approach for Automatic Number Plate Recognition”, in proc. of International Conference on Intelligent Systems and Signal Processing, pp 375-380, 2013.
- [6] D Renuka devi, D kanagapushpavalli “Automatic License Plate Recognition”, in proc. Of 3<sup>rd</sup> International Conference on Trendz in Information Science and Computing (TISC), pp 75-78, 2011.
- [7] Priyanto Hidayatullah, Nurjannah Syakrani, Ida Suhartini, Wildan Muhlis “Optical Character Recognition Improvement for License Plate Recognition in Indonesia”, UK Sim-AMSS 6<sup>th</sup> European Modelling Symposium, 2012.
- [8] Shoab Rehman Soomro, Mohammad Arslan Javed, Fahad Ahmed Menon “Vehicle Number Recognition System for automatic toll tax collection”, In proc. of International Conference of Robotics and Artificial Intelligence, pp 125-129, 2012.
- [9] Madhuri Latha.G, Chakravarthy.G “An Improved Bernsen Algorithm Approaches for License Plate Recognition”, *IOSR Journal of Electronics and Communication Engineering*, Vol.3, No.4, pp 01-05, 2012.
- [10] Phalgun Pandya, Mandeep Singh “Morphology Based Approach to recognize Number Plates in India”, *International Journal of soft computing and engineering*, Vol.1, No. 3, pp 107-110, 2011.