*International Journal of Information Technology and Knowledge Management*
*July-December 2008, Volume 1, No. 2, pp. 311-325*

# LICENSE PLATE RECOGNITION SYSTEM FOR INDIAN VEHICLES

SHISHIR KUMAR, SHASHANK AGARWAL & KUMAR SAURABH

ABSTRACT

This work deals with problematic from field of image processing and machine vision in construction of a license plate recognition system for Indian vehicles (LPRSIV). This problematic includes mathematical principles and algorithms, which ensure a process of number plate normalization, processes of proper characters segmentation, normalization and recognition. Work comparatively deals with methods achieving invariance of systems towards image skew using homography and various plate conditions. Work also contains few snapshots as example which will help in understanding the system and its functions.

***Keywords:*** Projection, Skew Correction, Homograph, Stroke, Gamma Correction, Correlation

## 1. INTRODUCTION

Massive integration of information technologies into all aspects of modern life in India caused demand for processing vehicles as conceptual resources in information systems. In India we have various kinds of number plates. Old number plate vehicles following 1939 series as well as the vehicles following the latest number plate format are seen on Indian lanes. The number plates following the new format can be of lengths 8, 9 or 10. As the number of vehicles entering the Indian lanes are increasing each day, so an automated number plate recognition system is becoming a necessity for tracking these vehicles. Number plate recognition systems can today be used in various traffic and security applications, such as parking, access and border control, or tracking of stolen cars. Automatic number plate recognition systems can be used in access control. A sample of the Indian License Plate is given in the figure below.

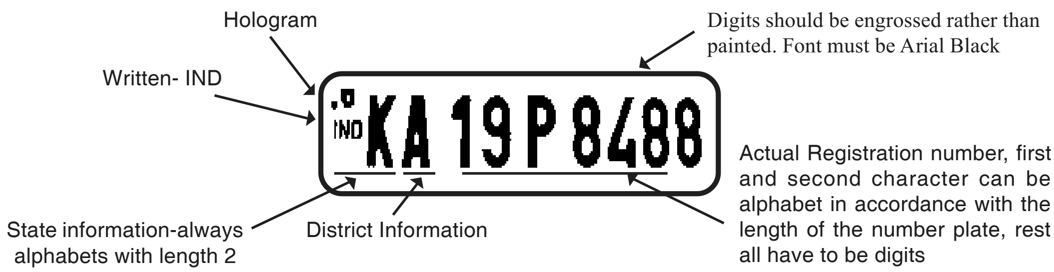



**Figure 1: A Sample of Indian Number Plate**

### 1.1 Some Mathematical Aspects

In most cases, vehicles are identified by their number plates, which are easily readable for humans, but not for machines. For machine, a number plate is only an rgb or gray picture as the case may be. Because of this, it is necessary to design robust mathematical machinery, which will be able to extract semantics from spatial domain of the captured image. These functions are implemented in so-called ANPR systems, where the acronym LPRSIV stands for an License Plate Recognition System for Indian Vehicles. LPRSIV system means transformation of data between the real environment and information systems.

It is assumed that the number plate is localized and the four corner points required for de-skewing an image are identified.

## 2. SKEW CORRECTION USING HOMOGRAPHY

The very first step in this problematic is to remove the skew from the sample image, so that the projections can be use for segmenting the image. This mechanism is a geometric operation over an image *f*. *x*, *y*.

### 2.1 Homography

Homography is a concept in the mathematical science of geometry. It is defined as a relation between two figures, such that any given point in one figure corresponds to one and only one point in the other, and vice versa. Here it used to obtain a skew corrected image from the original image. In the field of computer vision, a homography is defined in 2 dimensional space as a mapping between a point on a ground plane as seen from one camera, to the same point on the ground plane as seen from a second camera. This has many practical applications, most notably it provides a method for compositing 2D or 3D objects into an image or video with the correct pose.

### 2.2. Computation a Homography for Skew Correction

We are given 2D to 2D point correspondences $x_{i <-> } x'_i$ (these are points in $\mathcal{P}_2$ and hence are homogeneous vectors of size 3 × 1), and we have to find the homography $H$ (3 × 3 matrix) such that $x'_i = Hx_i$.

Note that $x'_i$ and $Hx_i$ are not numerically equal and they can differ by a scale factor. However, they have the same direction, and, hence $x'_i \times Hx_i = 0$.

Writing the $\boldsymbol{j}^{th}$ row of $H$ as $h^{jT}$, we have

$$Hx_i = \begin{bmatrix} h^{1^T} x_i \\ h^{2^T} x_i \\ h^{3^T} x_i \end{bmatrix}$$

Writing $x_i'^T = (x_{i,}' \, y_{i,} \, w_i')^T$, the cross product becomes:

$$x_1' \times Hx_i = \begin{bmatrix} y_i' h^{3^T} x_i - w_i' h^{2^T} x_i \\ w_i' h^{1^T} x_i - x_i' h^{3^T} x_i \\ x_i' h^{2^T} x_i - y_i' h^{1^T} x_i \end{bmatrix}$$

Since, $\quad h^{j^T} x_i = x_i^T h^j$

the system of equations $\quad x_i' \times Hx_i = 0$

can be written in terms of the unknowns (the entries of H) as:

$$\begin{bmatrix} 0^T & -w_i' x_i^T & y_i' x_i^T \\ w_i' x_i^T & 0^T & -x_i' x_i^T \\ -y_i' x_i^T & x_i' x_j^T & 0^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0$$

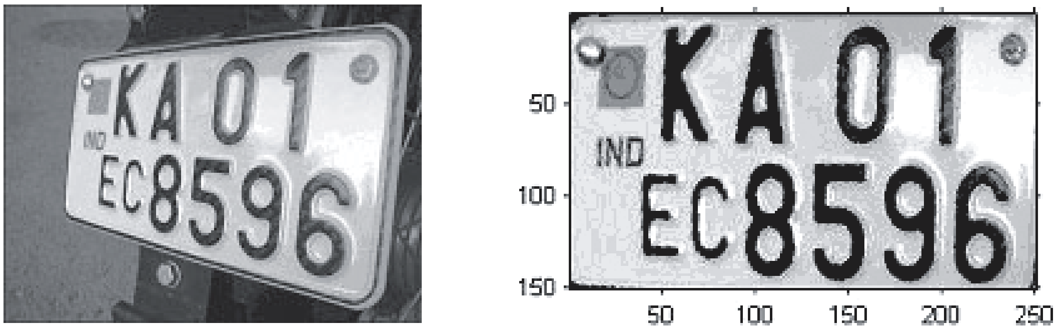
These equations have the form $A_i h = 0$ where $A_i$ is a 3 × 9 matrix and $h$ is a 9 × 1 vector (the entries of $H$). Note that $A_i$ has rank of 2 (third row is obtained, up to a scale, by a sum of $x_i'$ times the first row and $y_i'$ times the second), and, consequently, for each point correspondence we have really only two equations. We may choose to work with only the first two, but it doesn't harm to keep all three. It may be useful to keep all three equations because if $w_i' = 0$ (a point at infinity), then the first two collapse to a single equation.

Stacking up the equations for $i = 1, 2, 3, 4$ (four points) we have $Ah = 0$ where $A$ is a 12×9 matrix whose rank is 8 (of-course, you will not choose four points such that any three are collinear). Consequently $A$ has a 1-dimensional null space which provides a solution for $h$. Such a solution can only be determined up to a non-zero scale factor, which suits you fine because $H$ is anyway defined only up to a scale! A scale may be arbitrarily chosen for $h$ by insisting that $||h|| = 1$.

One can, of-course, stack up more equations by taking more point correspondences. The resulting over-determined system $Ah = 0$ may not have a solution at all (inconsistent measurements?). We can still find a least-squares solution: *minimize* $||Ah||$ *subject to* $||h||=1$.

In either case $h$ is given by the last column of $V$ where $A = U\Sigma V^T$ is the *singular value decomposition (SVD)* of $A$.

After the calculation of the value of $h$ the new image containing the area of interest, which can be further used for the identification process. An example of Skew Correction is shown in the figure [3].




**(a)** **(b)**

**Figure 2: (a) The Original Image Taken (b) Image After Skew Correction**

### 3. IDENTIFICATION OF THE NUMBER PLATE BACKGROUND

In India we do have rules regarding the background of the number plates, *i.e.* on the basis of the background of the number plate we can distinguish among the vehicles. The rules that are followed in India are enlisted as under:

- White background pertains to private vehicles
- Yellow background pertains to commercial vehicles
- Blue background pertains to foreign vehicles
- Red background pertains to vehicles of Indian President and the vehicles of Governors of different states (exception: these vehicles do not have any registration number, instead it carries a national emblem on it)
- Black background with numbers painted in yellow pertains to the embassy vehicles.

Also the number plate format for foreign vehicles is different from the format of Indian number plates. So if the categorization is done earlier, then at the later stages then a lot of information about the vehicle can be retrieved on this basis.

### 3.1. Background Identification

The background color of the image has to found out and then we can define the category of the vehicle. To identify the color of the background the rgb values for each pixel is found out and compared against a pre-defined range corresponding to the different possible colors of the image background. Each contribution adds to that particular color. After processing the complete image the color that has the maximum count is identified as the background of the number plate. Also it adds an extra information which makes the search set of the vehicle smaller.

For instance we have to locate a vehicle. If we have the license number of the vehicle and along with its type, the number of vehicles to be searched will reduce.

Some of the number plate patterns are shown below


(a) (b) (c) (d)

**Figure 3: (a) Foreign Number Plate (b) Commercial Number Plate (c) Private Number Plate (d) Old Govt. Number Plate**

## 4. NORMALIZATION OF THE NUMBER PLATE

The next step after obtaining a skew corrected image is the normalization of the obtained image. It is a very important process in the number plate recognition process because all the further steps rely on this process. Pre-processing on the image improves its quality and hence the recognition process becomes easier. If the image contains a lot of noise and does not have a proper contrast, then the segmentation of number plate would become difficult. Hence improving the image quality of image is very necessary [1,5].

In the process of image enhancement the pixel values of the image are changed. The values of the pixels before and after processing will be denoted by $r$ and $s$ respectively. These values are related by an expression of the form $s = T(r)$, where $T$ is the transformation that maps a pixel value $r$ into pixel value s. since we will be dealing with digital quantities, values of the transformation function typically are stored in one dimensional array.

### 4.1. Gamma Correction and Thresholding

Gamma correction, gamma nonlinearity, gamma encoding, or often simply gamma, is the name of a nonlinear operation used to code and decode luminance or tri-stimulus values in video or still image systems [7]. Gamma correction is, in the simplest cases, defined by the following **power-law** expression: $V_{out} = V^{\gamma}_{in}$ where the input and output values are non-negative real values, typically in a predetermined range such as 0 to 1. A gamma value $\gamma < 1$ is sometimes called an **encoding gamma**, and the process of encoding with this compressive power-law nonlinearity is called **gamma compression**; conversely a gamma value $\gamma > 1$ is called a **decoding gamma** and the application of the expansive power-law nonlinearity is called **gamma expansion.**

A variety of devices used for image capture, printing and display respond according to a power law. By convention, the exponent in the power law equation is referred to as ***gamma***. The process is used to correct this power-law response phenomenon is called ***gamma correction***.

After the gamma correction the image is converted into a grayscale. Then the contrast of the image is adjusted (*imadjust* can be used for this process in case of matlab). The intensity variation is found for the image (*strchlim* can be used to find the intensity variation in case of matlab) and then can be adjusted accordingly.

After completing all the above processes, adaptive thresholding is applied. We obtain a black and white image with some stroke and also some unwanted pieces. These would create a hindrance in the segmentation process. So some pre-processing have to be applied in order to remove these pieces.


**Figure 4: (a) Input Image (b) Image after Gamma Correction and Intensity Adjustment**

### 4.2. Removal of Unwanted Patterns

For removing the unwanted pieces, we select a threshold that corresponds to the average size of the character. If the size of the connected component is less then the threshold value, then it is identified as the unwanted pattern and is removed from the image [2].

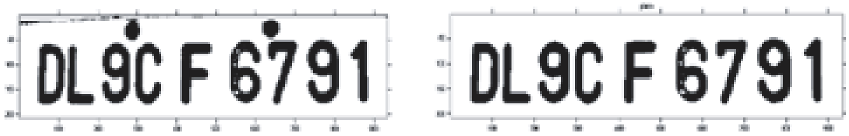



**Figure 5: (a) Image with Unwanted Patterns (b) Image After the Removal of Unwanted Patterns**

### 4.3. Stroke Deletion

Stroke deletion is very useful for mainly two reasons.The first one is if the stroke is not connected to characters it underlines, it may be considered as another character. The second reason is that it may cause errors on page segmentation, such as line segmentation and more particularly character segmentation if it is connected to characters. Actually, some individual characters can become touching characters with a stroke [2].

Also, some degradation is added. The RLE (Run-Length Encoding) algorithm, usually useful in transmission error correction, is used here to detect them. On a given row of pixels, consecutive ON-pixels sequences are detected and saved. First components are already defined in previous steps. Therefore we had thought about their mean width and height. All strokes with a width larger than three times the mean width and a height smaller than the mean height is considered as a stroke to remove [13]. Moreover, if the place where the stroke was detected remains a place with a large component, the object will be removed because of not convenient sizes comparing to characters sizes in general [3].

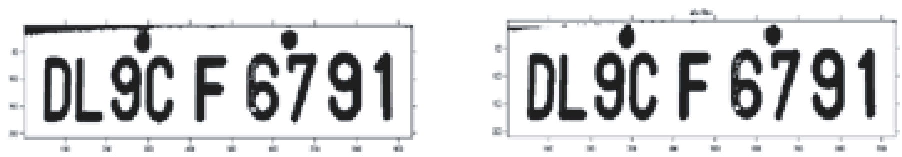



**Figure 6: (a) Image Before Stroke Deletion (b) Image After Stroke Deletion**

## 5. SEGMENTATION OF NUMBER PLATE

The next step after obtaining performing thresholding on the number plate is segmentation of the plate. This is important process because on the basis of these segments only the further processes can be carried out. If the segmentation does not works, a character can be improperly divided into two pieces, or two characters can be improperly merged together [1,8].

We can use a projection of a number plate for the segmentation. If we assume only one-row plates, the segmentation is a process of finding horizontal boundaries between characters whereas in the case of Indian number plates, we have both single row as well as two row number plates. So both vertical as well as horizontal projection will be useful on this case. Vertical projection will be used to segment the different rows in the license plate while horizontal projection is used to segment out each character.

The second phase of the segmentation is an enhancement of segments. The segment of a plate contains besides the character also undesirable elements such as dots and stretches as well as redundant space on the sides of character. There is a need to eliminate these elements and extract only the character.

### 5.1. Horizontal and Vertical Projection for Segmentation

After thresholding we first compute the vertical projection $f(x)$ to separate out the different rows and then the horizontal projection $g(x)$ is computed to separate the individual characters. We use vertical projection to find the boundaries between two rows and horizontal projection to find boundaries between the characters [1,9,12]. These boundaries correspond to the peak of the graphs of horizontal as well as vertical projection. (Figure 7.1, 7.2)

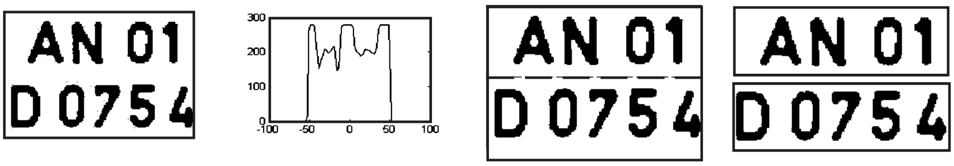



**Figure 7 (a) Number Plate after the Application of Adaptive Thresholding (b) Vertical Projection Graph Depicting the Peaks as the Places of Segmentation (c) Dotted Line Depicts the Line of Segmentation (d) The Two Separated Rows of Two Row Number Plate**

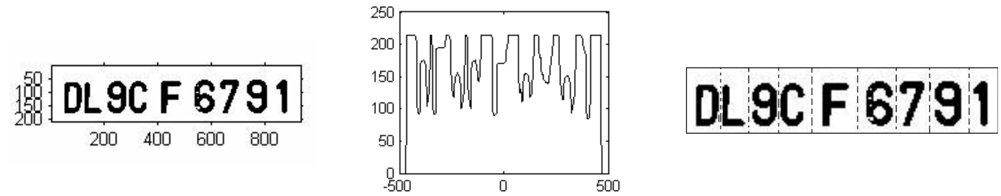



**Figure 8: (a) Number Plate After the Application of Adaptive Thresholding (b) Horizontal Projection Graph Depicting the Peaks as the Places of Segmentation for Obtaining the Character Segments (c) The Various Character Segments into which the Plate will be Segmented**

The segmentation algorithm is used to find the peaks of the projection graph, which corresponds to the spaces between two rows or the spaces between two characters as the case may be. For this, we need to define certain values corresponding to the horizontal $f(x)$ and vertical $g(x)$ projection graph.

- $v_m$ – The maximum value contained in the horizontal(or vertical) projection graph such as $v_m = \max(f(x))$ or $v_m = \max(g(x))$
- $v_a$ – The average value for horizontal and vertical projection graph
- $v_b$ - This value is used as a base for evaluation of peak height. The base value is always calculated as $v_b = 2\,v_a - v_m$. The $v_a$ must lie on vertical axis between the values $v_b$ and $v_m$.

The algorithm of segmentation iteratively finds the maximum peak in the graph of vertical or horizontal projection. The peak is treated as a space between characters in case of horizontal projection and space between two rows in the case of vertical projection, if it meets some additional conditions, such as height of peak. The algorithm then zeroizes the peak and iteratively repeats this process until no further space is found. This principle can be illustrated by the following steps:

(1) Determine the index of the maximum value of horizontal projection:

$x_m = \arg\max\{f(x)\}$ or $x_m = \arg\max\{g(x)\}$

(2) Detect the left and right foot of the peak as:

$x_l = \max\{x| f(x) <= c_x.f(x_m)\}$ or $x_l = \max\{x|g(x) <= c_x.g(x_m)\}$

$x_r = \min\{x| f(x) <= c_x.f(x_m)\}$ or $x_r = \min\{x|g(x) <= c_x.g(x_m)\}$

(3) Zeroize the horizontal projection $f(x)$ or vertical projection $g(x)$ on interval, $<x_l,x_r>$

(4) If $f(x_m) < c_w . v_m$ or $f(x_m) < c_w . v_m$, go to step 7.

(5) Divide the plate horizontally in the point $x_m$.

(6) Go to step 1.

(7) End.

Two different constants have been used in the algorithm above. The constant $c_x$ is used to determine foots of peak $x_m$. The optimal value of $c_x$ is 0.7.

The constant $c_w$ determines the minimum height of the peak related to the maximum value of the projection ($v_m$). If the height of the peak is below this minimum, the peak will not be considered as a space between characters. It is important to choose a value of constant $c_w$ carefully. An inadequate small value causes that too many peaks will be treated as spaces, and characters will be improperly divided. A big value of $c_w$ causes that not all regular peaks will be treated as spaces, and characters will be improperly merged together. The optimal value of $c_w$ is 0.86. To ensure a proper behavior of the algorithm, constants $c_x$ and $c_w$ should meet the following condition:

$$V(x_l, x_m, x_r) \bullet \mathrm{P}: c_w.v_m > f(x_1) \wedge f(x_r)$$
$$V(x_l, x_m, x_r) \bullet P: c_w.v_m > g(x_1) \wedge g(x_r)$$

Where, $P$ is a set of all detected peaks $m\ x$ with corresponding foots $x_l$ and $x_r$.

## 5.2. Removal of Unwanted Patterns from Segments

Each horizontal segment of plate may contain besides the character also the redundant space and other undesirable elements. Single the segment has been pr-processed it only contains black and white pixels. The neighboring pixels are grouped together into a larger piece, and one of them is character. Our task is to divide the segment into various pieces and keep only that piece which represents a regular character. The concept is illustrated in the figure.

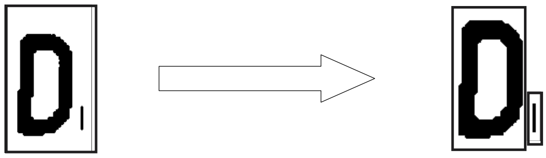


**Figure 9: Horizontal Segment of Number Plates Contains Various Unwanted Pieces Needs have to be Removed**

### 5.3. Piece Extraction and Heuristic Analysis

Let the segment be defined by a discrete function $f(x, y)$ in the relative co-ordinate system such as (0, 0) for the upper left corner and ($w$–1, $h$–1) for the bottom right corner of the segment, where $w$ and $h$ are the width and the height of the segment respectively.

A piece $P$ is a set of all neighboring pixels $(x, y)$, which represents the continuous element. A pixel $(x, y)$ belong to piece $P$ if there is at least one pixel $(x', y')$ from $P$ such that $(x, y)$ and $(x', y')$ are neighbors [6, 9].

***Algorithm***

- Find all the connected components in the piece using connected components method. We use 8-connected pixels to find the connected pieces.
- The size of each connected component is found out.
- **Heuristic Analysis:** The piece that constitutes the maximum size is selected and rest are left out
- The selected piece is sent to OCR for the character recognition

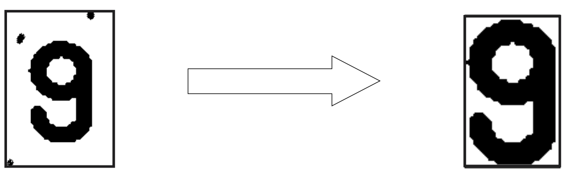


**Figure 10: The Various Pieces of in the Horizontal Segments are Removed and the Segment having the Largest Size is Taken**

## 6. CHARACTER RECOGNITION

After obtaining the individual patterns to be identified, each pattern is sent to the OCR (Optical Character Recognition) System. Here the nearest match is found. The pattern whose correlation is nearest to a particular sample alphabet is labeled as that respective character [4,10].

### 6.1. Optical Character Recognition

We use Optical Character Recognition to identify each of the separated segments. The steps followed in character recognition process are as follows:

***Algorithm***

- The segment received for identification is first clipped in such a way that the four corners of the new segment co-inside with the four possible corners of the pattern to be identified.
- The clipped pattern is now resized to the sample pattern size (i.e. the size of the pattern with which the correlation is to be found).
- Now the correlation of this segment and the sample character patterns is found.
- The sample pattern that corresponds to the maximum correlation is accepted, and the pattern which was to be identified is labeled as that particular character.
- Each identified letter is written into a text file.
- The above process is repeated until all the segments are identified.

### 6.2. Semantic Analysis of Patterns

In the case of Indian number plates the number plate length can be 8, 9 or 10. The formats corresponding to the enlisted as follows:

- If the number of segments counts to 7:
  - The number plate follows 1939 series which says first three characters will be alphabets and rest will be numbers.
- If the number of segments counts to 8:
  - The first two characters will be alphabets and rest of them will be numbers.
- If the number of segments counts to 9:
  - The first, second and fifth characters to be identified will be alphabets while the rest will be numbers
- If the number of segments counts to 10:

- The first, second, fifth and sixth characters will be alphabets and the rest will be numbers

- EXCEPTION:
  - For the case of vehicles of Delhi not following 1939 series the fourth character is an alphabet that contains the information about the type of vehicle it is.

For the case of foreign vehicles the rules followed are:

- The third and fourth characters will be alphabets and the rest will be numbers

Using the above semantics the comparison process can be reduced i.e. on the basis of the number of segment it can be found whether that particular character will be an alphabet or a number. Hence we can restrict our search using the above semantics. Also, the results obtained will be better as the correlation is found with only a particular type.

Still there are certain alphabets which might have been identified incorrectly. For example, in case of ‘D’, ‘Q’, and ‘O’ if the condition of the number plate is very bad, in those cases the correlation found might not give the correct results. For those cases we use the method of symmetry to find the exact alphabet. For instance, if we have identified a character as ‘D’, ‘Q’ or ‘O’ then we divide the character horizontally or vertically as the case requires and try to find the symmetry of two halves. For example, if we are trying to verify a character as ‘D’ then we segment it horizontally and find the symmetry of the two halves. If they are identical then it’s a ‘D’ else not. Similarly in the case of ‘O’ we can divide it horizontally or vertically since it is symmetrical both ways. Such heuristics improve the results to a great extent.

## 7. SYNTACTICAL ANALYSIS OF THE IDENTIFIED NUMBER

After completing all the above processes, we finally obtain a license plate number. But in some cases it may be that the identified number is faulty. These errors have to be rectified using some heuristics. Also, after obtaining the license number the state to which the vehicle belongs can be found by comparing the initials. Some additional information can also be retrieved from the license number. For instance, in case of Delhi vehicles with the help of fourth character the type of the vehicle can be found out **[11, 14]**. The list of characters that defines the type of vehicles in case of Delhi is as follows:

| *Character* | *Vehicle Type* |
|---|---|
| P | Public Vehicles and Buses |
| C | Cars and SUV’s |
| S | Two Wheelers |
| L | Lorry |
| R | Auto Rikshaw |
| Y | Hire Vehicles |

### 7.1. Information Retrieval from License Plate

Using the above list, we can extract some additional information from the identified number plate. Also, we know that the initials of none of the states start with '*I*' or '*X*' and so on. So we can take some action accordingly to rectify the number in case the number plate was faulty. At the end, whole of the information is written into a text file and saved.

## 8. DISPLAYING THE RESULTS

At last when the whole number is recognized and the type of vehicle and state related information is retrieved the output of the text file is reformatted and saved in the following format.

- License number
- State or Union territory to which the vehicle belongs. In case of 1939 series, the number plate of vehicle is labeled as Old number plate
- "State" or "Union territory" as the case may be
- Type of vehicle for the case of vehicles pertaining to Delhi. For the vehicles of other states NA is saved depicting Not Applicable
- The type of vehicle which was found on the basis of the background color of the number plate

After appending all the above information into the text file it is saved and finally the results are displayed.

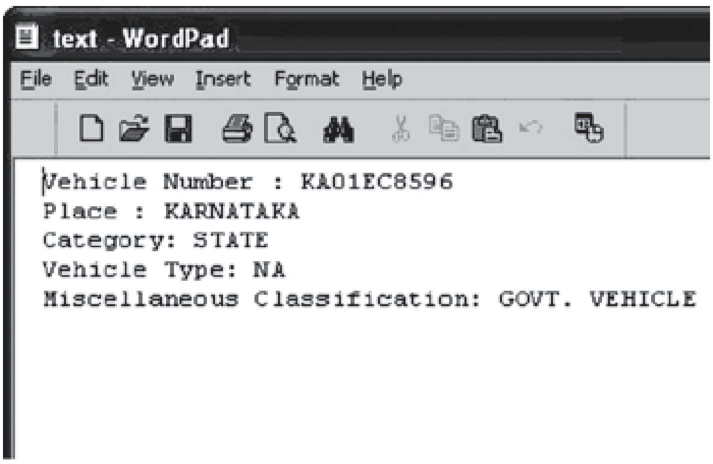



**Figure 11: Format in which the Result is Displayed**

## 9. RESULTS

The tests were made on images taken from various sources like internet, scanned images etc. Some of the samples were made with the help of Paint and other softwares. At the later stages real images were taken with the help of digital camera, inside and outside the campus.

Of all the images taken the ones that are supported by the system, 90% of the images give us accurate results. 5% cases give partially successful results and in the remaining 5% cases the results were totally incorrect.For images that are not supported by the system, 25% of images give a correct result. 50% images give partially correct results whereas 25% cases showed incorrect results.

Although the system works properly in most of the cases but if the image is too blurred or the contrast of image is very poor, then it may not work properly. Also, we have taken an assumption that the input license plate is already localized.

In the case of skew correction the four boundary points of the image to be de-skewed have to be given manually. Finding the boundary points is not yet dealt. If the number plate contains unnecessary designs or the font of the number plate changes regularly, it creates a problem. The vehicles that do not follow the standard rules of Indian Number Plates pose a problem to the system. There are many number plates in the vehicles whose condition is so bad that even a human eye cannot recognize it. In those cases the system is obviously unable to obtain the expected results.

There are many areas on which we not worked yet, due to the 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. Also, in the case of skew correction, the boundary points of the image, from where the image has to be de-skewed needs to be given manually. So, this can be another module that we can deal with, in the future. Work can also be done on correction of images containing motion blurring and blurring due to the use of low resolution camera.

The segmentation of characters that are connected to each other and the characters which are too close to each other poses a problem to the system. Its rectification can taken as a part of future enhancement. We are currently using OCR for character recognition. The character recognition process can also be done by PNN (Probabilistic Neural Network). This can also be taken as a possibility of the enhancements for the project.

## REFERENCES

[1] Ondrej Martinsky: Algorithmic And Mathematical Principles of Automatic Number Plate Recognition Systems, Faculty of Information Technology Department of Intelligent Systems, Brno University of Technology, (2007).

[2] Céline THILLOU: Degraded Character Recognition, Dipl ome d ´Etudes Approfondies en Sciences Appliquées 2003-2004

[3] *http://www.cse.iitd.ernet.in/~suban/vision/geometry/node24.html*

[4] Diego Barragan Guerrero:Optical Character Recognition (OCR), 2007. Available: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=18169&objectType=file

[5] Zhang Y., Zhang C.: New Algorithm for Character Segmentation of License Plate, Intelligent Vehicles Symposium, *IEEE*, (2003).

[6] Wikipedia: Automatic Number Plate Recognition *http://en.wikipedia.org/wiki/* Automatic_number_plate_recognition

[7] Rafael C. Gonzalez & Richard E. Woods: Digital Image Processing, Second Edition

[8] J. R. Parker, Pavol Federl: An Approach to License Plate Recognition, University of Calgary.

[9] Gao, D. and Zhou, J.: Car License Plates Detection from Complex Sense, *International Conference on Signal Processing*.

[10] C.N. Anagnostopoulos, I. Anagnostopoulos, V. Loumos and E. Kayafas (2008) : Alicense Plate Recognition Algorithm for *Intelligent Transportaion System application*.

[11] F. Martín, M. García, J. L. Alba, (2002, Jun.), New Methods for Automatic Reading of VLP's (Vehicle License Plates), presented at IASTED *International Conference Signal Processing, Pattern Recognition, and Applications*, SPPRA 2002. [Online]. Available: *http://www.gpi.tsc.uvigo.es/pub/papers/sppra02.pdf*.

[12] W.-Y. Chen and S.-Y. Chen, Adaptative Page Segmentation Forcolor Technical Journals Cover Images, *Image and Vision Computing*, **16**, (1998), 855–877.

[13] Shigueo Nomura, Keiji Yamanaka, Osamu Katai, Hiroshi Kawakami, Takayuki Shiose, A Novel Adaptive Morphological Approach for Degraded Character Image Segmentation, Pattern Recognition, (2002, Nov.).

[14] Jesse Hansen: A Matlab Project in Optical Character Recognition (OCR).

**Shishir Kumar, Shashank Agarwal & Kumar Saurabh**
Dept. of Computer Science & Engineering
Jaypee Institute of Engineering & Technology
Guna (MP) India
E-mail: *dr.shishir@yahoo.com*