ISSN 2047-3338

ANPRSRE: Automatic Number Plate Recognition System for Real-Time Environments

M.J. Qureshi[1], K.H. Asif[2] and Khalid Bashir[3]

[1,2]University of Engineering and Technology, Lahore-Pakistan

[3]School of Agriculture and Resource Economics, University of Western Australia

***Abstract*— The purpose of ANPRSRE 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 or at the entrance of an office gate. ANPRSRE system, based on regular PC with video camera, catches video frames which include a visible car license no. plate and processes them. Once a license no. plate is detected, its digits are recognized, displayed on the GUI monitor and then checked against a database. The focus is on the design of algorithms used for extracting the license plate from a single image, isolating the characters of the plate and identifying the individual characters. There have been similar past project. The purpose of ANPRSRE project was first and foremost to improve the accuracy of the program, and whenever possible its time-complexity. This project have good accuracy according to the tests we made on the set of images, recorded videos and were successful and it finds out license no. plates from an video/image and recognize its number.**

***Index Terms*— Real-time, Image Processing, Time-complexity and Identification**

I. INTRODUCTION

Automatic Number/License Plate Recognition plays an important role in numerous applications such as unattended parking lots, security control of restricted areas, traffic law enforcement and automatic toll collection. Number/License plates may have arbitrary sizes, orientations and positions. And, if complex backgrounds are involved, detecting license plates can become quite a challenge.

Typically, an ANPRSRE process consists of two main stages: i) locating license plates and ii) identifying license numbers. In the first stage, license plate candidates are determined based on the features of license plates. Features commonly employed have been derived from the license plate format and the alphanumeric characters constituting license numbers. ANPRSRE is a method that uses optical character recognition to read the numbers on license plate/number plate on vehicles.

It can use existing closed-circuit television (CCTV) camera or general video camera. It can use by police as a method of monitoring traffic activity or at an office parking area etc. ANPRSRE can be used to store the images captured by the cameras as well as the text from the license plate, with some configurable to store a photograph of the driver.

ANPRSRE technology tends to be region specific, owing to plate variation from place to place. The software aspect of the system runs on standard PC hardware and can be linked to databases. It first uses a series of image manipulation techniques [1] [2] to detect, normalize and enhance the image of the number plate, and finally optical character recognition to extract the alphanumeric of the license plate. ANPRSRE system is deployed the entire process to be performed at the location in real-time. When done, the information captured of the plate alphanumeric, date-time and any other information that is required is completed.

II. SCOPE OF THE PROJECT

Scope of the project includes:

- Keep track of all the vehicles passed from that location.
- Read the license number of the vehicle.
- Time of vehicle with its number is also stored.
- It can read number of vehicle from front side and as well as from rear of the vehicle.
- Find out the number of the vehicle from image, from recorded movie and as well as from live video capturing device.
- This software can be implemented at different places like car parking or at an organization entry gate.
- This software will help to keep record of incoming/outgoing vehicles.
- Wanted cars can be tracked by police.

III. PROPOSED METHODOLOGY

For number plate recognition to occur, two process need to be performed. Firstly, the number-plate must be found in the picture, and then the characters on the number-plate must be read. As this area is one of great commercial interest, there are no published methods for this area.

Therefore it can be regarded as the most challenging aspect of number plate recognition. From various literature sources, there appears to be two methods that can be adopted for locating a number plate. The first of these methods is based on threshold the image so that there is a contrast between the number plate characters and the background. The image is

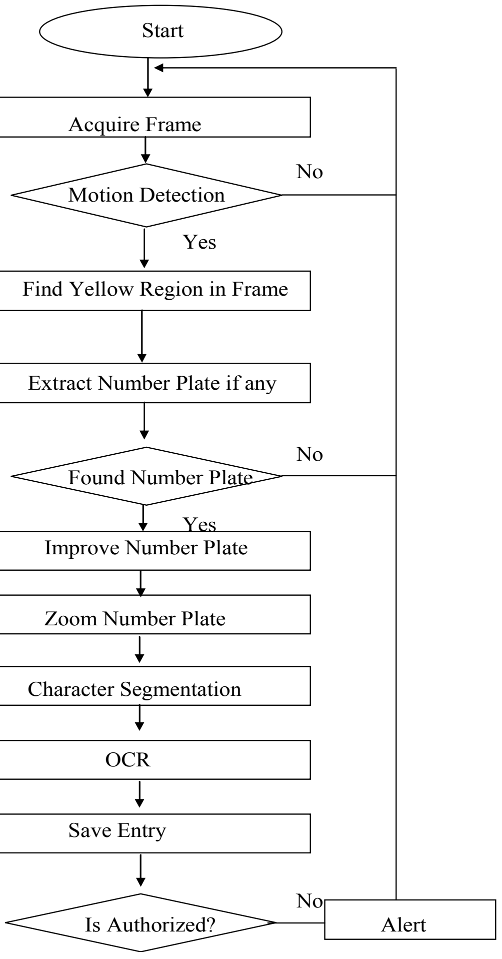

Fig. 1. Flow diagram of ANPRSRE system

then searched for regions containing several blobs, roughly the size of number-plate characters, upon a contrasting background, and the second method is to utilize neural networks.
Here remains the flow diagram of ANPRSRE system which shows the flow of the system (Fig. 1):

A. HSV Image

HSV (Hue, Saturation, and Value) is one of several color system used to select colors from a color wheel or plate. This color system is considerably closer than the RGB system to the way in which human experience and described color sensation. In artist terminology, hue, saturation and value refer to tint, shade and tone.

The HSV color space is formulated by looking at the RGB color cube along its gray axis, which results in the hexagonally shaped color palette shown in Fig. 2.

Hue is expressed as an angle around a color hexagon. Value is measured along the axis of the cone. Saturation (purity of color) is measured as the distance from the axis.

The HSV color system is based on cylindrical coordinates. Converting from RGB to HSV is simply the development of equation.

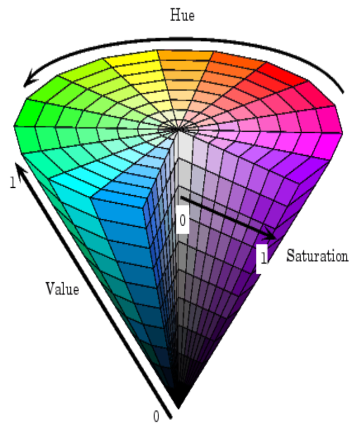

Fig. 2. HSV color cone

B. Edge Detection

Find edges in an intensity image. Edge takes an intensity image *I* as its input, and returns a binary image *BW* of the same size as *I*, with 1's where the function finds edges in *I* and 0's elsewhere.

Edge supports six different edge-finding methods:

- The Sobel method finds edges using the Sobel approximation to the derivative. It returns edges at those points where the gradient of I is maximum.
- The Prewitt method finds edges using the Prewitt approximation to the derivative. It returns edges at those points where the gradient of I is maximum.
- The Roberts method finds edges using the Roberts approximation to the derivative. It returns edges at those points where the gradient of I is maximum.
- The Laplacian of Gaussian method finds edges by looking for zero crossings after filtering I with a Laplacian of Gaussian filter.
- The zero-cross method finds edges by looking for zero crossings after filtering I with a filter you specify.
- The Canny method finds edges by looking for local maxima of the gradient of I. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be "fooled" by noise, and more likely to detect true weak edges.

C. Labeling the Image

Matlab [4] function 'bwlabel' label connected components in a binary image.

L = bwlabel (BW, n) returns a matrix L, of the same size as BW, containing labels for the connected objects in BW. n can have a value of either 4 or 8, where 4 specifies 4-connected objects and 8 specifies 8-connected objects; if the argument is omitted, it defaults to 8.

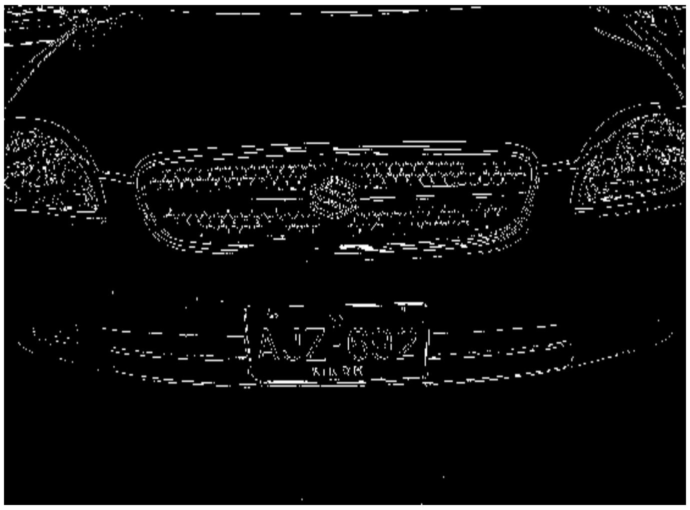

Fig. 3. Edge detection using Sobel Method

The elements of L are integer values greater than or equal to 0. The pixels labeled 0 are the background. The pixels labeled 1 make up one object; the pixels labeled 2 make up a second object, and so on.

For instance, if we label the Fig. 4 then it becomes as shown in Fig. 5.

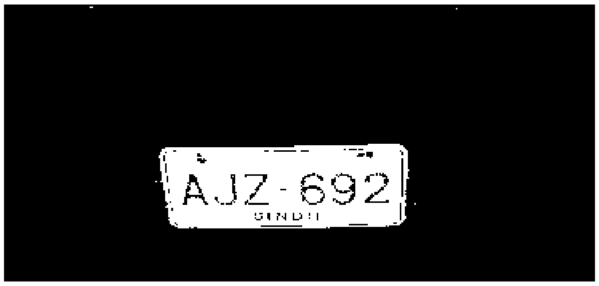

Fig. 4. A logical picture showing number plate

After labeling the image it becomes as shown in Fig. 5 all the regions having value 0 means black becomes white and all the regions having value 1 means white in Fig. 4 becomes color as shown in Fig. 5.

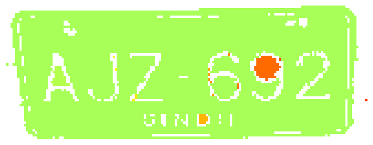

Fig. 5. A labeled image of number plate

D. Connected Components

This example illustrates using 4-connected objects:

```
BW = [1     1     1     0     0     0
0     0
      1     1     1     0     1     1
0     0
      1     1     1     0     1     1
0     0
      1     1     1     0     0     0
1     0
      1     1     1     0     0     0
1     0
      1     1     1     0     0     0
1     0
      1     1     1     0     0     1
1     0
      1     1     1     0     0     0
0     0];

L = bwlabel(BW,4)

L =

     1     1     1     0     0     0
0     0
     1     1     1     0     2     2
0     0
     1     1     1     0     2     2
0     0
     1     1     1     0     0     0
3     0
     1     1     1     0     0     0
3     0
     1     1     1     0     0     0
3     0
     1     1     1     0     0     3
3     0
     1     1     1     0     0     0
0     0


[r,c] = find (L==2);
rc = [r c]

rc =

     2     5
     3     5
     2     6
     3     6
```

E. Features of an Image

Matlab function ‘regionprops’ computes a set of measurements for each labeled region in the label matrix L. Positive integer elements of L correspond to different regions. For example, the set of elements of L equal to 1 corresponds

Table 1: Set of valid measurement strings

'Area'	'Image'	'EulerNumber'
'Centroid'	'FilledImage'	'Extrema'
'BoundingBox'	'FilledArea'	'EquivDiameter'
'MajorAxisLength'	'ConvexHull'	'Solidity'
'MinorAxisLength'	'ConvexImage'	'Extent'
'Eccentricity'	'ConvexArea'	'PixelList'
'Orientation'		

to region 1; the set of elements of L equal to 2 corresponds to region 2; and so on. Its syntax is as follows:

stats = regionprops(L, measurements), stats is a structure array of length max(L(:)). The fields of the structure array denote different measurements for each region, as specified by *measurements*.

measurements can be a comma-separated list of strings, a cell array containing strings, the single string 'all', or the single string 'basic'.

The set of valid measurement strings includes the following (Table 1):

In Table 1, `'Area'` - Scalar; the actual number of pixels in the region, (This value may differ slightly from the value returned by `'bwarea'`, which weights different patterns of pixels differently), `'BoundingBox'` - 1-by-4 vector; the smallest rectangle that can contain the region. The format of the vector is `[x y width height]`, where `x` and `y` are the *x*- and *y*-coordinates of the upper-left corner of the rectangle, and `width` and `height` are the width and height of the rectangle. Note that `x` and `y` are always non-integer values, because they are the spatial coordinates for the upper-left corner of a pixel in the image; for example, if this pixel is the third pixel in the fifth row of the image, then `x` = 2.5 and `y` = 4.5.

Fig. 6 illustrates the centroid and bounding box. The region consists of the white pixels; the green box is the bounding box, and the red dot is the centroid.

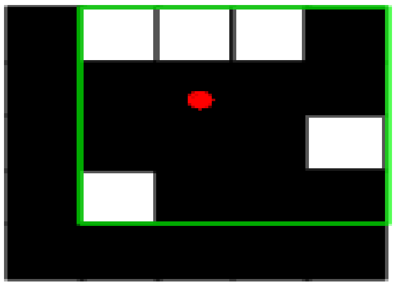

Fig. 6. Describing bounding box

F. Radon Transform

RADON Compute Radon transforms. The RADON function computes the Radon transform, which is the projection of the image intensity along a radial line oriented at a specific angle. RADON returns the Radon transform of the intensity image for the angle theta degrees. If theta is a scalar, the result R is a column vector containing the Radon transforms for theta degrees. If theta is a vector, then R is a matrix in which each column is the Radon transform for one of the angles in theta. If we omit theta, it defaults to 0:179. It can be of class double, logical or of any integer class. All other inputs and outputs are of class double. The radial coordinates returned in Xp are the values along the x-prime axis, which is oriented at theta degrees counterclockwise from the x-axis.

For instance, the number plate shown in Fig. 7, its angle can be found through radon transform and then it can be rotated according to the x-axis at the found angle.

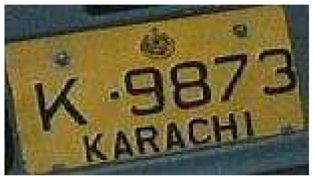

Fig. 7. Number plate not straight

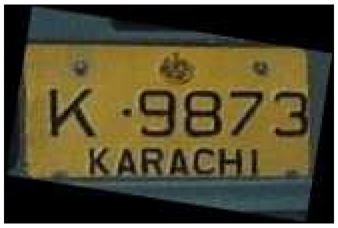

Fig. 8. Rotated number plate

Now this number plate can be further improved as shown in Fig. 9.

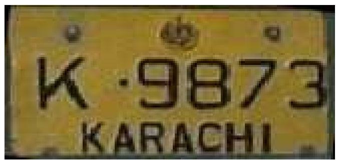

Fig. 9. Improved number plate

The purpose of doing all of this activity was to ease the task of Optical Character Recognition.

G. Row Vector

A digital image [3] [5] can be represented as a Matlab matrix:

$$f = \begin{pmatrix} f(1,1) & f(1,2) & f(1,3) & \ldots & f(1,N) \\ f(2,1) & f(2,2) & f(2,3) & \ldots & f(2,N) \\ \ldots & \ldots & \ldots & & \ldots \\ f(M,1) & f(M,2) & f(M,3) & \ldots & f(M,N) \end{pmatrix}$$

The notation f (p, q) denotes the element located in row p and column q. For example, f (6, 2) is the element in 6th row and 2nd column of the matrix. A 1 x N matrix is called a *row vector matrix.* Row vector matrix has only 1 row but having N number of columns.

H. Column Vector

A (M x 1) matrix is called a *column vector matrix*. Column vector matrix has only 1 column but have M number of columns.

I. Template Matching

This process involves the use of a database of characters or templates. There exists a template for all possible input characters. For recognition to occur, the current input character is compared to each template to find either an exact match, or the template with the closest representation of the input character.

If *I (x, y)* is the input character, *Tn (x, y)* is the template *n*, then the matching function *s (I, Tn)* will return a value indicating how well template *n* matches the input character.

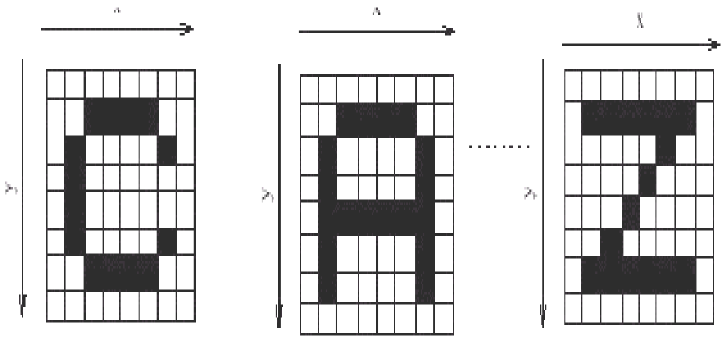

Fig. 10. Template matching

Some of the more common matching functions are:

City block

$$s(I, T_n) = \sum_{i=0}^{w}\sum_{j=0}^{h} |I(i,j) - T_n(i,j)|$$

Euclidean

$$s(I, T_n) = \sum_{i=0}^{w}\sum_{j=0}^{h} (I(i,j) - T_n(i,j))^2$$

Cross Correlation

$$s(I, T_n) = \sum_{i=0}^{w}\sum_{j=0}^{h} I(i,j)\, T_n(i,j)$$

Normalised Correlation

$$s(I, T_n) = \frac{\sum_{i=0}^{w}\sum_{j=0}^{h} (I(i,j) - |I|)\,(T_n(i,j) - |T_n|)}{\sqrt{\sum_{i=0}^{w}\sum_{j=0}^{h} (I(i,j) - |I|)^2}\ \sqrt{\sum_{i=0}^{w}\sum_{j=0}^{h} (T_n(i,j) - |T_n|)^2}}$$

Character recognition is achieved by identifying which Tn gives the best value of matching function, s (I, Tn). The method can only be successful if the input character can the stored templates are of the same or similar font. Template matching can be performed on binary, threshold characters or on gray-level characters. For gray-level characters, it is more common for Normalized Correlation to be used as this provides improved resistance to variations in brightness and contrast between the input character and the stored template.

- *Correlation:* is quite simple in principle. Given an image *f(x, y),* the correlation problem is to find all places in the image that match a given sub image (also called mask or template) *w(x, y).* One approach for finding matches is to treat *w(x, y)* as a spatial filter and compute the sum of products for each location of *w* in *f.* Then the best match of *w(x, y)* in *f(x, y)* is the location of the maximum values in the resulting correlation image. Unless *w(x, y)* is small, the approach just described generally becomes *computationally* intensive.

Through this method of template matching system can distinguish between different characters as shown in Fig. 11.

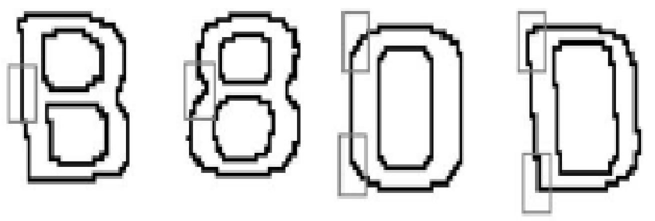

Fig. 11. Distinguish points of ambiguous characters

IV. CONCLUSIONS

The methodology we have used in this work involves two different fields of study: i) image processing [2] and ii) artificial neural networks. In this research we have studied the different features regarding license plate format including shape, height-to-width ratio, noise, broken number plate and color. We also explored the character features including line, blob, and the sign transition of gradient magnitudes, the aspect ratio of characters, the distribution of intervals between characters, and the alignment of characters. There were two major tasks involved in the identification stage, character separation and character recognition. Character separation has been accomplished by such technique as relaxation labeling, connected components. There has been a large number of character recognition techniques reported. They include genetic algorithms, artificial neural networks and finite automata. In this study, we pay more attention on accuracy and time complexity.

REFERENCES

[1]. Fang, Y., Masaki, I., and Horn, B.K.P., "Depth-Based Target Segmentation for Intelligent Vehicles: Fusion of Radar and Binocular Stereo", IEEE Transactions on Intelligent Transportation Systems, 3(3) pp. 196--202, September, 2002.

[2]. Carlo Ratti and Paul Richens, "Urban Texture Analysis with Image Processing Techniques", Proceedings of the Eighth International Conference on Computer Aided Architectural Design Futures, pp. 49-64, Atlanta, June 1999.

[3]. Jain A K, 1989, “Fundamentals of Digital Image Processing”, (Prentice-Hall, London).

[4]. Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins, “Digital image processing using MATLAB”.

[5]. Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, 2nd Edition.

[6]. www.apian-tech.com\products\anpr_overview.html

[7]. www.anpr.org.uk