www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 8 August, 2013 Page No. 2412-2414

Computational Algorithm for Automatic Recognition of Vehicle Registration Plates

Ms. Preetika Sharma[1], Mr Gaurav Sapra[2], Mr Rachit Agarwal[3]

[1]*Assistant Professor , Electrical & Electronics Department,,*
UIET, Panjab University, Sector 25 ,Chandigarh
preetikamadhav@yahoo.co.in

[2]*Assistant Professor, Electrical & Electronics Department,,*
UIET, Panjab University, Sector 25 ,Chandigarh
Gaurav_sapra2001@yahoo.com

[3] *Student, Electrical & Electronics Department,*
UIET, Panjab University, Sector 25 ,Chandigarh
Contact.rachit@gmail.com

Abstract: ***This paper aims to describe a computational algorithm for automatic recognition of vehicle registration plate from recorded video. There is extensive research and literature on different ways to implement automatic recognition of vehicle registration plate. Popular methods involve the use of technologies like computer vision, image precession, artificial neural networks and fuzzy logic systems. Some of these have been implemented, but none so far have been able to guarantee high levels of accuracy. Also, most of these systems work on still images captured from a camera. In this approach, we begin by obtaining frames having image of moving car from the video using motion detection. The region containing the number plate is then identified by the Canny edge detection algorithm and Hough transformation. Segmentation of the plate is then undertaken, using histogram analysis, to obtain the separate characters. Neural techniques are then applied to recognize the individual characters. With further work on the components of such a system and their successful integration, a largely automated system can be developed for traffic and law enforcement purposes. The deployment of such a system in different states or countries carries with it its own unique issues, and there is a need to configure the system to properly function in the required domain.***

Keywords: ***Edge Detection, Hough Transformation Segmentation, Extraction, Character Recognition*.**

1. Introduction

In recent years, there has been a marked increase in the usage of surveillance equipment like CCTV cameras in public areas and traffic cameras at major intersections. In most cases, there is a requirement for continuous monitoring of the output by human eyes. However, with simple image processing techniques, it is possible to drastically reduce the need for human intervention. Intelligent algorithms applied to video feeds can be used to simplify many processes that may be slowed down by human limitations of speed and accuracy.

Massive integration of information technologies into all aspects of modern life has caused demand for processing vehicles as conceptual resources in information systems. This can be achieved by special intelligent equipment which is be able to recognize vehicles by their number plates in a real environment and reflect it into conceptual resources. Because of this, various recognition techniques have been developed and number plate recognition systems are today used in various traffic and security applications.

In parking, number plates can be used to calculate duration of the parking. The difference in entry and exit times can be used to calculate the parking fee. Automatic number plate recognition systems can also be used in access control to grant access only to vehicles of authorized personnel. In some countries, ANPR systems installed on country borders automatically detect and monitor border crossings. In traffic control, vehicles can be directed to different lanes for a better congestion control in busy urban communications during the rush hours.

The majority of vehicle registration plate recognition systems in common use throughout the world are designed for specific situations, including fixed position of the vehicle in the image and plate on the vehicle, stationary vehicles, colours of the plate background and characters, etc. Assumptions are also usually made regarding the lighting conditions and the amount of noise present on the plate. [1]

This paper describes a smart and simple algorithm for a generic vehicle license plate recognition system from video. The input video was initially separated into different frames at regular time intervals. Suitable frames were identified by means of motion detection, and were scanned for the license plate of the moving vehicle. This was accomplished using edge detection techniques and Hough transformation for detection of straight lines edges. The obtained plate region was then separated into characters, which were recognized by the use of the concepts of neural networks.

2. Frame Extraction

A video acquisition device, like high definition digital camera, captures a stream of image frames. The video may be acquired at variable angles and lighting conditions. The appropriate frame for further processing is acquired using the concepts of computer vision. Consecutive frames are compared and if sufficient difference is noted, it is assumed that a moving vehicle is present in that image, and the frame is selected for further processing. The product of this module is a usable image containing predominantly the vehicle whose registration plate is to be identified.

3. Number Plate Detection

Number plate detection from the extracted frames involves the following steps:

3.1 Pre-Processing

Basic image processing techniques are applied to generate a sufficient quality of image for further processing. The color image is converted to a gray-scale image and its histogram is equalized to stretch the contrast to the full range, which is then normalized.
If r_k is the intensity of the k^{th} pixel, than the equalized intensity S_k can be computed as equation1:

$$S_k = T(r_k) = \sum_{j=1}^{k} p_r(r_j) = \sum_{j=1}^{k} \frac{n_j}{n} \qquad (1)$$

Where T is the equalization function, $p_r(r_j)$ is the probability density of intensity j, n_j is the number of pixels having intensity j and n is the total number of pixels.

3.2 Edge Detection

Canny Edge Detection algorithm[2] is used to determine the edges in the image. In this algorithm, first the image is subjected to Gaussian filtering for noise reduction via image smoothening. The resultant image G (x,y) can be computed as equation 2:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} \qquad (2)$$

Where σ is the standard deviation, and (x,y) are the pixel coordinates.
Gradients at each pixel in the smoothed image are determined by applying what is known as the Sobel-operator. The gradients in the x and y directions respectively are approximated by applying the kernels:

$$K_{GX} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad (3)$$

$$K_{GY} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad (4)$$

Then gradients are computed, which represent the change in intensities. The gradient magnitudes (also known as the edge strengths) can be determined as a Euclidean distance measure by applying the law of Pythagoras. It is sometimes simplified by applying Manhattan distance measure to reduce the computational complexity.

$$|G| = \sqrt{G^2{}_x + G^2{}_y} \qquad (5)$$

$$|G| = |G_x| + |G_y| \qquad (6)$$

The direction of the edges can be determined as:

$$\theta = \arctan\left(\frac{|G_y|}{|G_x|}\right) \qquad (7)$$

Non-maximal suppression is used to convert the "blurred" edges in the image of the gradient magnitudes to "sharp" edges. Basically this is done by preserving all local maxima in the gradient image, and deleting everything else. The gradient direction θ is rounded to the nearest 45°, corresponding to the use of an 8-connected neighbourhood. The edge strength of the current pixel is compared with the edge strength of the pixel in the positive and negative gradient direction. i.e. if the gradient direction is, compare with the pixels to the north and south. If the edge strength of the current pixel is largest; the value of the edge strength is preserved else it is removed.
Edge thresholding (hysteresis), using a high and low threshold, gives the final edges. Edge pixels stronger than the high threshold are marked as strong; edge pixels weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked as weak. Strong edges are interpreted as "certain edges", and can immediately be included in the final edge image. Weak edges are included if and only if they are connected to strong edges.[8]

3.3 Straight Line Detection

The Hough Transformation Algorithm [9] is used to select straight lines from these edges. The straight line equation in polar coordinates is used.

$$r(\theta) = x_0 \cos\theta + y_0 \sin\theta \qquad (8)$$

The points whose plots intersect in the (r,θ) planes are collinear. The straight lines having perpendicular distance from the origin and slope, within range for being approximately horizontal or vertical and in a region of interest, are selected.

3.4 Plate Extraction

To detect a rectangular number plate from the output of the Hough Transform two methods may be used – (i) comparing the aspect ratio of all the possible rectangles in the Hough Transform output with that expected for a number plate or (ii) morphological dilation techniques .[5]

4. Segmentation

Firstly, the cut-out image of the license plate is deskewed so that further processing is made simpler. The fundamental problem of this mechanism is to determine an angle, under which the plate is skewed (or sheared). Then, ‘
deskewing of the evaluated plate can be realized by a trivial affine transformation.

The next step after the deskewing of the number plate area is the segmentation of the plate. Segmentation is one of the most important processes in the automatic number plate recognition, because all further steps rely on it. If the segmentation fails, a character may be improperly divided into two pieces, or two characters may be improperly merged together. We may use a horizontal projection of a number plate for the segmentation, or one of the more sophisticated methods, such as segmentation using the neural networks. If we assume only one-row plates, the segmentation is a process of finding horizontal boundaries between characters.

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. Each segment is analysed and the different unconnected regions of the segment are considered. The region most likely to be the character is cut out from the segment, by comparing characteristics like size and percentage of black pixels in the region [12].

5. Character Recognition [11]

Prior to the recognition algorithm, the characters are normalized. Normalization is to refine the characters into a block containing no extra white spaces (pixels) in all the four sides of the characters. Normalization also involves ensuring uniformity in the brightness and contrast characteristics of the segments. Then each character is fit to equal size for matching the characters with the database; input images must be equal-sized with the database characters. The next step is template matching. The character image is compared with the ones in the database and the best similarity is measured. To measure the similarity and find the best match, neural network principles are applied. Additionally, heuristic analysis is undertaken to avoid false detections due to noise or irrelevant marks on the plate. The character recognition is configured to the needs of the particular state of country where the application is to be deployed. This involves syntactical analysis based on characteristic features of the number-assigning system of that region.

References

[1] Badr, M.M. Abdelwahab, A.M Thabet and A. M. Abdelsadek, “Automatic Number Plate Recognition System”, Annals of the University of Craiova, Mathematics and Computer Science Series Volume 38(1), 2011, Pages 62-71.

[2] V Abolghasemi and Alireza Ahmadyfard, “An edge-based color-aided method for license plate detection”, Image and Vision Computing Volume 27, Issue 8, 2 July 2009, Pages 1134–1142

[3] C.N.E.Anagnostopoulos, I.E. Anagnostopoulos, I.D. Psoroulas, V. Loumos and E. Kayafas, “License Plate Recognition from Still Images and Video Sequences: A Survey”, IEEE Transactions on Intelligent Transportation Systems 9 (2008), no. 3, 377–391.

[4] O. Martinsky, “Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems”, 2007

[5] M. S. Nixon, A. S. Aguado, “Feature Extraction and Image Processing”, Reed Educational and Professional Publishing, 2002.

[6] R. C. Gonzalez, R. E. Woods, “Digital Image Processing”, Addison Wesley, 1992.

[7] K. Fukunaga, “Introduction to statistical pattern recognition”, Academic Press, 1990

[8] J. Canny, “A Computational Approach to Edge Detection”, IEEE transactions on pattern analysis and machine intelligence, 1986.

[9] P.E. Hart, R. O. Duda, “ Use of the Hough Transformation to Detect Lines and Curves in Pictures”, 1971.

[10] Shapiro, D. Dimov, S. Bonchev, V. Velichkov, G. Gluhchev, “Adaptive License Plate Image Extraction”, International Conference Computer Systems and Technologies, 2004.

[11] Y. Zhang, C. Zhang, “New Algorithm for Character Segmentation of License Plate”, IEEE Intelligent Vehicles Symposium, 2003.

[12] J. Wang, J. Jean, “Segmentation of merged characters by neural networks and shortest-path”, Symposium on Applied Computing, 1993.

[13] P. Smagt, “Comparative study of neural network algorithms applied to optical character recognition”, International conference on Industrial and engineering applications of artificial intelligence and expert systems, 1990.