**License Plate Recognition System based on Image Processing Using Labview**

Kuldeepak
Department of Electronics Engg.,
YMCA UST,
Faridabad, Haryana, India

Monika kaushik
Department of Electronics Engg.,
GIU,
Hisar, Haryana, India

Munish Vashisht,
Department of Electronics Engg.,
YMCA UST,
Faridabad, Haryana, India

Abstract—A license plate recognition (LPR) system is one kind of an intelligent transport system and is of considerable interest because of its potential applications in highway electronic toll collection and traffic monitoring systems. A lot of work has been done regarding LPR systems for Korean, Chinese, European and US license plates that generated many commercial products. However, little work has been done for Indian license plate recognition systems. The purpose of this paper is to develop a real time application which recognizes license plates from cars at a gate, for example at the entrance of a parking area or a border crossing. The system, based on regular PC with video camera, catches video frames which include a visible car license plate and processes them. Once a license plate is detected, its digits are recognized, displayed on the user interface or 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. The proposed system has been implemented using Vision Assistant 8.2.1 & Labview 11.0 the recognition of about 98% vehicles shows that the system is quite efficient.

Keywords - Image Acquisition; License Plate Extraction; Segmentation; Recognition.

I. INTRODUCTION

License plate recognition (LPR) is an image-processing technology used to identify vehicles by their license plates. This technology is gaining popularity in security and traffic installations. License plate recognition system is an application of computer vision. Computer vision is a process of using a computer to extract high level information from a digital image. The lack of standardization among different license plates (i.e., the dimension and the layout of the license plates). LPR system in lieu of the following four stages: image acquisition, license plate extraction and license plate segmentation and license plate recognition phases.

A. Image Acquisition

Image Acquisition is the first step in an LPR system and there are a number of ways to acquire images, the current literature discusses different image acquisition methods used by various authors. Yan et. al. [19] used an image acquisition card that converts video signals to digital images based on some hardware-based image pre-processing. Naito et. al. [13], [14],[16] developed a sensing system, which uses two CCDs (Charge Coupled Devices) and a prism to split an incident ray into two lights with different intensities. The main feature of this sensing system is that it covers wide illumination conditions from twilight to noon under sunshine, and this system is capable of capturing images of fast moving vehicles without blurring. Salgado et. al. [15] used a Sensor subsystem having a high resolution CCD camera supplemented with a number of new digital operation capabilities. Kim et. al. [17] uses a video camera to acquire the image. Comelli et. al. [6] used a TV camera and a frame grabber card to acquire the image for the developed vehicle LPR system.

B. License Plate Extraction

License plate extraction is the most important phase in an LPR system. This section discusses some of the previous work done during the extraction phase. Hontani et. al. [20] proposed a method for extracting characters without prior knowledge of their position and size in the image. The technique is based on scale shape analysis, which is the position of the figure in the scale t indicates the inherent characteristic size of corresponding figures. All these characteristic points enable the extraction of the figure from the given image that has line-type shapes locally and blob-type shapes globally. In the scale shape analysis, Gaussian filters at various scales blur the given image and larger size shapes appear at larger scales. To detect these scales the idea of principal curvature plane is introduced. By means of normalized principal curvatures, characteristic points are extracted from the scale space x-y-t. The position (x, y) indicates the position of the figure and the scale t indicates the inherent characteristic size of corresponding figures. These points enable the extraction of the figure from the given image that has line-type shapes locally and blob-type shapes globally. Kim et. al. [17] used two Neural Network-based filters and a post processor to combine two filtered images in order to locate the license plates. The two Neural Networks used are vertical and horizontal filters, which examine small windows of vertical and horizontal cross sections of an image and decide whether each window contains a license plate. Cross-sections have sufficient information for distinguishing a plate from the background. Lee et. al. [5] and Park et. al. [11] devised a method to extract Korean license plate depending on the color of the plate. A Korean license plate is composed of two different colors, one for characters and other for background and depending on this they are divided into three categories. In this method a neural network is used for extracting color of a pixel by HLS (Hue, Lightness...
and Saturation) values of eight neighbouring pixels and a node of maximum value is chosen as a representative color. After every pixel of input image is converted into one of the four groups, horizontal and vertical histogram of white, red and green (i.e. Korean plates contains white, red and green colors) are calculated to extract a plate region. To select a probable plate region horizontal to vertical ratio of plate is used. Dong et. al [10] presented histogram based approach for the extraction phase. Kim G. M [9] used Hough transform for the extraction of the license plate. The algorithm behind the method consists of five steps. The first step is to threshold the gray scale source image, which leads to a binary image. Then in the second stage the resulting image is passed through two parallel sequences, in order to extract horizontal and vertical line segments respectively. The result is an image with edges highlighted. In the third step the resultant image is then used as input to the Hough transform, this produces a list of lines in the form of accumulator cells. In fourth step, the above cells are then analyzed and line segments are computed. Finally the list of horizontal and vertical line segments is combined and any rectangular regions matching the dimensions of a license plate are kept as candidate regions. The disadvantage is that, this method requires huge memory and is computationally expensive.

C. Segmentation:

This section discusses previous work done for the segmentation of characters. Many different approaches have been proposed in the literature and some of them are as follows, Nieuwoudt et. al. [8] used region growing for segmentation of characters. The basic idea behind region growing is to identify one or more criteria that are characteristic for the desired region. After establishing the criteria, the image is searched for any pixels that fulfill the requirements. Whenever such a pixel is encountered, its neighbours are checked, and if any of the neighbours also match the criteria, both the pixels are considered as belonging to the same region. Morel et. al. [7] used partial differential equations (PDE) based technique; Neural network and fuzzy logic were adopted in for segmentation into individual characters.

D. Recognition:

This section presents the methods that were used to classify and then recognize the individual characters. The classification is based on the extracted features. These features are then classified using either the statistical, syntactic or neural approaches. Some of the previous work in the classification and recognition of characters is as follows, Hasen et. al. [22] discusses a statistical pattern recognition approach for recognition but their technique found to be inefficient. This approach is based on the probabilistic model and uses statistical pattern recognition approach. Cowell et. al. [23] discussed the recognition of individual Arabic and Latin characters. Their approach identifies the characters based on the number of black pixel rows and columns of the character and comparison of those values to a set of templates or signatures in the database. Cowell et. al. [21] discusses the thinning of Arabic characters to extract essential structural information of each character which may be later used for the classification stage. Mei Yu et. al. [18] and Naito et. al. [12] used template matching. Template matching involves the use of a database of characters or templates. There is a separate template for each possible input character. Recognition is achieved by comparing the current input character to each of template in order to find the one which matches the best. If I(x,y) is the input character, Tn(x,y) is template n, then the matching function s(I,Tn) will return a value indicating how well template n matches the input. Hamami et. al. [24] adopted a structural or syntactic approach to recognize characters in a text document, this technique can yield a better result when applied on the recognition of individual characters. This approach is based on the detection of holes and concavities in the four directions (up, down, left and right), which permits the classification of characters into different classes. In addition, secondary characteristics are used in order to differentiate between the characters of each class. The approaches discussed in this paragraph are based on the structural information of the characters and uses syntactic pattern recognition approach. Hu [1] proposed seven moment that can be used as features to classify the characters. These moments are invariant to scaling, rotation and translation. The obtained moments acts as the features, which are passed to the neural network for the classification or recognition of characters. Zernike moments both the rotation variant and rotation invariant features can be extracted. These features then uses neural network for the recognition phase. Neural network accepts any set of distinguishable features of a pattern as input. It then trains the network using the input data and the training algorithms to recognize the input pattern (In this case character).

II. IMAGE ACQUISITION

Complete the following steps to acquire images.

1. Click Start » Programs » National Instruments » Vision Assistant 8.2.1.
2. Click Acquire Image in the Welcome screen to view the Acquisition functions. If Vision Assistant is already running, click the Acquire Image button in the toolbar. We must have one of the following device and driver software combinations to acquire live images in Vision Assistant.
   - National Instruments IMAQ device and NI-MAQ 2.5 or later.
   - IEEE 1394 industrial camera and NI-MAQ for IEEE 1394 Cameras 1.5 or later

Click Acquire Image. The Parameter window displays the IMAQ devices and channels installed on the computer.

Snapping an image:

1. Click File » Acquire Image.
2. Click Acquire Image in the Acquisition function list.
3. Select the appropriate device and channel.
4. Click the Acquire Single Image button to acquire a single image with the IMAQ device and display it.
5. Click the Store Acquired Image in Browser button to send the image to the Image Browser.
6. Click Close to exit the Parameter window.

III. SCRIPT DEVELOPMENT

A. Extracting color planes from image:

Since the color information is redundant so we extract color plane from the acquired 32-bit color image to make it an 8-bit greyscale image.

B. Image mask:

An image mask isolates parts of an image for processing. If a function has an image mask parameter, the function process or analysis depends on both the source image and the image mask. An image mask is an 8-bit binary image that is the same size as or smaller than the inspection image. Pixels in the image mask determine whether corresponding pixels in the inspection image are processed.

C. Optical Character Recognition (OCR):

The exact mechanisms that allow humans to recognize objects are yet to be understood, but the three basic principles are already well known by scientists – integrity, purposefulness and adaptability. These principles constitute the core of OCR allowing it to replicate natural or human-like recognition. Optical Character Recognition provides machine vision functions we can use in an application to perform OCR. OCR is the process by which the machine vision software reads text and/or characters in an image.

The process of locating characters in an image is often referred to as character segmentation. Before we can train characters, we must set up OCR to determine the criteria that segment the characters you want to train. When we finish segmenting the characters, we’ll use OCR to train the characters, storing information that enables OCR to recognize the same characters in other images. We train the OCR software by providing a character value for each of the segmented characters, creating a unique representation of each segmented character. You then save the character set to a character set file to use later in an OCR reading procedure.

D. Reading Characters:

When we perform the reading procedure, the machine vision application we created with OCR functions segments each object in the image and compares it to characters in the character set you created during the training procedure. OCR extracts unique features from each segmented object in the image and compares each object to each character stored in the character set. OCR returns the objects that best match characters in the character set as the recognized characters.

E. Character Segmentation:

Character segmentation applies to both the training and reading procedures. Character segmentation refers to the process of locating and separating each character in the image from the background.
F. Thresholding:

Thresholding is one of the most important concepts in the segmentation process. Thresholding is separating image pixels into foreground and background pixels based on their intensity values. Foreground pixels are those whose intensity values are within the lower and upper threshold values of the threshold range. Background pixels are those whose intensity values lie outside the lower and upper threshold values of the threshold range. OCR includes one manual method and three automatic methods of calculating the thresholding range:

Fixed Range is a method by which you manually set the threshold value. This method processes gray scale images quickly, but requires that lighting remain uniform across the ROI and constant from image to image.

The following three automatic thresholding methods are affected by the pixel intensity of the objects in the ROI. If the objects are dark on a light background, the automatic methods calculate the high threshold value and set the low threshold value to the lower value of the threshold limits. If the objects are light on a dark background, the automatic methods calculate the low threshold value and set the high threshold value to the upper value of the threshold limits.

- Uniform is a method by which OCR calculates a single threshold value and uses that value to extract pixels from items across the entire ROI. This method is fast and is the best option when lighting remains uniform across the ROI.

- Linear is a method that divides the ROI into blocks, calculates different threshold values for the blocks on the left and right side of an ROI, and linearly interpolates values for the blocks in between. This method is useful when one side of the ROI is brighter than the other and the light intensity changes uniformly across the ROI.

- Non linear is a method that divides the ROI into blocks, calculates a threshold value for each block, and uses the resulting value to extract pixel data. OCR includes a method by which you can improve performance during automatic thresholding, which includes the Uniform, Linear, and Non linear methods.

Optimize for Speed allows us to determine if accuracy or speed takes precedence in the threshold calculation algorithm. If speed takes precedence, enable Optimize for Speed to perform the thresholding calculation more quickly, but less accurately. If accuracy takes precedence, disable Optimize for Speed to perform the thresholding calculation more slowly, but more accurately. If we enable Optimize for Speed, we also can enable Bi modal calculation to configure OCR to calculate both the lower and upper threshold levels for images that are dominated by two pixel intensity levels.

G. Threshold Limits:

Threshold limits are bounds on the value of the threshold calculated by the automatic threshold calculation algorithms. For example, if the threshold limits are 10 and 240, OCR uses only intensities between 10 and 240 as the threshold value. Use the threshold limits to prevent the OCR automatic threshold algorithms from returning too low or too high values for the threshold in a noisy image or an image that contains a low population of dark or light pixels. The default range is 0 to 255.
The software has been tested for different vehicle images. The results of some images are as under. The table below consist vehicle image, its correct number, number read by our system result.

V. CONCLUSION

The process of vehicle number plate recognition requires a very high degree of accuracy when we are working on a very busy road or parking which may not be possible manually as a human being tends to get fatigued due to monotonous nature of the job and they cannot keep track of the vehicles when there are multiple vehicles are passing in a very short time. To overcome this problem, many efforts have been made by the researchers across the globe for last many years. A similar effort has been made in this work to develop an accurate and automatic number plate recognition system. We have used Vision assistant 8.2.1 along with LabVIEW 11.0 to obtain the desired results. The setup has been tested for vehicles containing different number plates from different states. In the process of final evaluation after optimizing the parameters like brightness, contrast and gamma, adjustments, optimum values for lightening and the angle from which the image is to be taken. We get an overall efficiency of 98% for this system. Though this accuracy is not acceptable in general, but still the system can be used for vehicle identification. It may be concluded that the project
has been by and far successful. It can give us a relative advantage of data acquisition and online warning in case of stolen vehicles which is not possible by traditional man handled check posts. While thousands of vehicles pass in a day. Though we have achieved an accuracy of 98% by optimizing various parameters, it is required that for the task as sensitive as tracking stolen vehicles and monitoring vehicles for homeland security an accuracy of 100% cannot be compromised with. Therefore to achieve this, further optimization is required. Also, the issues like stains, smudges, blurred regions & different font style and sizes are need to be taken care of. This work can be further extended to minimize the errors due to them.

REFERENCES