RESEARCH ARTICLE OPEN ACCESS

Automated License Plate Recognition for Toll Booth Application

Ketan S. Shevale
(Department of Electronics and Telecommunication, SAOE, Pune University, Pune)

ABSTRACT
This paper describes the Smart Vehicle Screening System, which can be installed into a tollbooth for automated recognition of vehicle license plate information using a photograph of a vehicle. An automated system could then be implemented to control the payment of fees, parking areas, highways, bridges or tunnels, etc. There are considered an approach to identify vehicle through recognizing of it license plate using image fusion, neural networks and threshold techniques as well as some experimental results to recognize the license plate successfully.
Keywords – License plate recognition, real world application, pattern recognition, segmentation

I. INTRODUCTION

The task of vehicle identification can be solved by vehicle license plate recognition. It can be used in many applications such as entrance admission, security, parking control, airport or harbor cargo control, road traffic control, speed control and so on. A number of commercial software is developed in this area. However, they cannot be readily used when vehicle image is provided in different styles and formats. Proposed approach allows removing this drawback by ensemble of two methods: (i) detection and extraction of image region included license plate from source images flow and (ii) recognition of character presented on the license plate. Image processing techniques such as edge detection, thresholding and resembling have been used to locate and isolate the license plate and the characters. The neural network was used for successful recognition the license plate number. Once a license plate has been accurately identified, information about the vehicle can be obtained from various databases. Should the information suggest that there is anything suspicious about the vehicle, appropriate actions can be taken. The algorithm of license plate recognition (LPR) consists of the following steps: (i) to capture the car's images, (ii) to deblur of image frames, (iii) to extract image of license plate, (iv) to extract characters from license plate image, (v) to recognize license plate characters and identify the vehicle.

II. EXISTING SYSTEM

ANPR systems have been implemented in many countries like Australia, Korea and few others [1]. Strict implementation of license plate standards in these countries has helped the early development of ANPR systems. These systems use standard features of the license plates such as: dimensions of plate, border for the plate, color and font of characters, etc. help to localize the number plate easily and identify the license number of the vehicle. In India, number plate standards are rarely followed. Wide variations are found in terms of font types, script, size, placement and color of the number plates. In few cases, other unwanted decorations are present on the number plate. Also, unlike other countries, no special features are available on Indian number plates to ease their recognition process. Hence, currently only manual recording systems are used and ANPR has not been commercially implemented in India.

III. PROPOSED SYSTEM

In designing this system (Fig. 1), various Image Processing algorithms were designed in Matlab and implemented on the Digital Signal Processor TMS320DM6437 which is optimized for video and image processing applications. A rear image of a vehicle is captured and processed using various algorithms. Initially, the number plate area is localized using a novel „feature-based number plate localization method which consists of many algorithms. This algorithm satisfactorily eliminates all the background noise and preserves only the number plate area in the image. This area is then segmented into individual characters using Image Scissoring algorithm. After this step, the characters are extracted from the gray-sale image and each character is enhanced using some character enhancement techniques. These characters are given to the character recognition module, which uses statistical feature Extraction to recognize the characters.

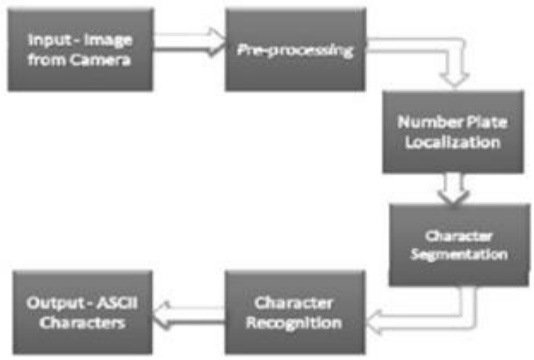

Fig. 1 System Architecture

IV. FUNDAMENTAL STEPS INVOLVED IN IMAGE PROCESSING

Image Acquisition-:

Image acquisition is the first process shown in Figure. To do so requires an imaging sensor and the capability to digitize the signal produced by the sensor. The sensor could be a monochrome or color TV camera that produces an entire image of the problem domain every 1/30 seconds. The imaging sensor could also be a line-scan camera that produces a single image line at a time. If the output of the camera or other imaging sensor is not already in digital form, an analog-to-digital converter digitizes it.

Image Enhancement-:

Image enhancement is among the simplest and most appealing areas of digital image -processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image.

Image Restoration-:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective

Color Image Processing-:

Color image processing is an area that has been gaining in importance because of the significant increase in the use of digital images over the Internet. Color is used as the basis for extracting features of interest in an image.

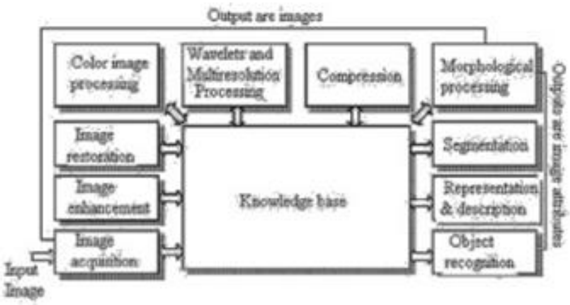

Fig. 2 Block Diagram of Image Processing

Wavelets and Multiresolution Processing -:

Wavelets are the foundation for representing images in various degrees of resolution. In particular, this is used for image data compression and for pyramidal representation, in which images are subdivided successively into smaller regions.

Compression-:

Compression deals with techniques for reducing the storage required to save an image, or the bandwidth required to transmit it

Morphological Processing-:

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape

Segmentation-:

Segmentation procedures partition an image into its constituent parts or objects. In general autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way towards successful solution of imaging problems that require objects to be identified individually. On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In terms of character recognition, the key role of segmentation is to extract individual characters and words from the background. The output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i. e., the set of pixels separating one image region from another) or all the points in the region itself. In either case, converting the data to a from suitable for computer processing is necessary.

Representation and Description-:

The next stage is Representation and description. Here, the first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing.

Object Recognition-:

Recognition is the process that assigns a label (e.g. “vehicle”) to an object based on its descriptors.

Knowledge base-:

Knowledge about a problem domain is coded into an image processing system in the form of a knowledge database. This knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information. These are the basic steps involved in image processing. The first step when vehicle breaks sensor then camera captures image and convert color image into gray scale image.

Fig. 3 Gray-scale image

Thresholding is the simplest method of image segmentation. From a grayscale image, Thresholding can be used to create binary images During the Thresholding process, individual pixels in an image are marked as "object" pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as "background" pixels otherwise. This convention is known as threshold above. Variants include threshold below, which is opposite of threshold above; threshold inside, where a pixel is labeled "object" if its value is between two thresholds; and threshold outside, which is the opposite of threshold inside. Typically, an object pixel is given a value of “1” while a background pixel is given a value of “0.” Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's labels. A Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. The equation of a Gaussian function in one dimension is,

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

In two dimensions, it is the product of two such Gaussians, one in each dimension:

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

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. When applied in two dimensions, this formula produces a surface whose contours are concentric circles with a Gaussian distribution from the center point. Values from this distribution are used to build a convolution matrix which is applied to the original image. Each pixel's new value is set to a weighted average of that pixel's neighborhood. The original pixel's value receives the heaviest weight (having the highest Gaussian value) and neighboring pixels receive smaller weights as their distance to the original pixel increases. This results in a blur that preserves boundaries and edges better than other, more uniform blurring filters.

The input gray-scale image (fig.3) is adaptively converted into binary image (fig. 4) using Ostus method. This method is better suited for our application compared to other Adaptive binarization methods like the Niblacks method. Using **Template matching** we can detecting license plate. Template matching is a technique in digital image processing for finding small parts of an image which match a Template image. It can be used in manufacturing as a part of quality control, a way to navigate a mobile robot, or as a way to detect edges in images.

Template-based matching and convolution :-

A basic method of template matching uses a convolution mask (template), tailored to a specific feature of the search image, which we want to detect. This technique can be easily performed on grey images or edge images. The convolution output will be highest at places where the image structure matches the mask structure, where large image values get multiplied by large mask values. This method is normally implemented by first picking out a part of the search image to use as a template: We will call the search image S(x, y), where (x, y) represent the coordinates of each pixel in the search image. We will call the template T(x t, y t), where (xt, yt) represent the coordinates of each pixel in the template. We then simply move the center (or the origin) of the template T(x t, y t) over each (x, y) point in the search image and calculate the sum of products between the coefficients in S(x, y) and T(xt, yt) over the whole area spanned by the template. As all possible positions of the template with respect to the search image are considered, the position with the highest score is the best position. This method is sometimes referred to as 'Linear Spatial Filtering' and thetemplate is called a filter mask. For example, one way to handle translation problems on images, using template matching is to compare the intensities of the pixels, using the SAD (Sum of absolute differences) measure. A pixel in the search image with coordinates (xs, ys) has intensity Is(xs, ys) and a pixel in the template with coordinates (xt, yt) has intensity It(xt, yt). Thus the absolute difference in the pixel intensities is defined as Diff(xs,ys, x t, y t) = | Is(xs, ys) – It(x t, y t) |.

The mathematical representation of the idea about looping through the pixels in the search image as we translate the origin of the template at every pixel and take the SAD measure is the following:

Srows and Scols denote the rows and the columns of the search image and Trows and Tcols denote the rows and the columns of the template image, respectively. In this method the lowest SAD score gives the estimate for the best position of template within the search image. The method is simple to implement and understand.

Step 1: A mask having shape of inverted „L and size equal to maximum possible character dimensions is rolled throughout the binary image. At every increment, a position is shortlisted as possible character location if: There is at least a single white pixel on the mask and there is at least a single white pixel on the immediate next row and column of the mask (fig 4).

Fig. 4 Output of step 1

Step 2: Size of each shortlisted character calculated. If it is less than half of maximum possible character size that location discarded (fig.5)
Step 3: White pixel density of each probable character is calculated. If it is above 40% of total number of pixels, only then the location is preserved (fig. 5). All the preceding steps are carried out in a single iteration to achieve time optimization.

Fig. 5 Output of step 2

Step 4: For a set of rows having height equal to maximum possible number plate height, white pixel density is calculated. If it is not above certain threshold, that area is discarded (fig. 6).

Fig. 6 Output of step 4

Step 5: For a set of columns having width equal to maximum possible number plate width, white pixel density is calculated. If it is not above certain threshold, that area is again discarded (fig. 7).

Fig. 7 Output of step 5

Step 6: Number of characters in the finalized number plate areas is calculated. If number of characters is less than four, then that area is discarded. If two number plate areas with nearly same number of characters are found in close vicinity of each other, then those areas are merged together. After applying these steps, the number plate within the image is exactly located and all other background noise is eliminated. Number plate is now extracted (fig. 8) from the input binary image and is then eroded using square of size 2X2 which eliminates overlapping of characters before segmentation.

Fig. 8 Output of step 6

Character Recognition
We use template matching for character recognition. We have a database which is having 3 to 4 types of different character fonts according to Indian license plate. We scan license plate when we get 1 value that identifies characters start because our license plate is black and white. At that pixel we use template matching for character recognition. Store all identified characters in local variable check that string in our toll booth database. If vehicle pays toll then the car is allowed to go and if not paid then pay the toll first and then allow.

V. CONCLUSION

The system works satisfactorily for wide variations in illumination conditions and different types of number plates commonly found in India. It is definitely a better alternative to the existing manual systems in India.

REFERENCES

[1] Leonard G. C. Hamey, Colin Priest, "*Automatic Number Plate Recognition for Australian Conditions*", Proceedings of the Digital Imaging Computing: Techniques and Applications (DICTA), pp. 14- 21, December 2005.

[2] *Indian Central Motor Vehicle Act, Bear Act*, Rule no.49, 50.

[3] Mohamed El-Adawi, Hesham Abd el Moneim Keshk, Mona Mahmoud Haragi, "*Automatic License Plate Recognition*", IEEE Transactions on Intelligent Transport Systems, vol. 5, pp. 42- 53, March 2004.

[4] Charl Coetzee, C. Botha, C. Weber, D. Trier, "*PC Based Number Plate Recognition System*", Proceedings of IEEE International Symposium on Industrial Electronics, vol. 2, pp. 605 - 610, July 1998.

[5] Mohamed El-Adawi, Hesham Abd el Moneim Keshk, Mona Mahmoud Haragi, "*Automatic License Plate Recognition*",IEEE Transactions on Intelligent Transport Systems, vol. 5,pp. 42-53, March 2004.

[6] Yo-Ping Huang, Shi-Yong Lai,Wei-Po Chuang, "*A Template- Based Model for License Plate Recognition*", IEEE International Conference on Networking, Sensing & Control, March 21-23, 200