Int. Computer Symposium, Dec. 15-17, 2004, Taipei, Taiwan.

A Practical License Plate Recognition System on PDAs

Yo-Ping Huang and Tienwei Tsai
Department of Computer Science and Engineering
Tatung University
Taipei, Taiwan 10451 R.O.C.
E-mail: yphuang@ttu.edu.tw

Abstract- *Vehicle license plate recognition systems have long been expected to be applied to traffic surveillance and monitoring, such as finding stolen cars, controlling access to parking lots and gathering traffic flow information. There are two common considerations in those systems: accuracy and real-time response. In our proposed electronic billing (E-Bill) system, however, we focus mainly on its portability and still keep acceptable accuracy. The E-Bill system is best suited to be used in the parallel parking spaces or the parking lots without access control. It provides an excellent framework for parking clerks to recognize the license plate through a PDA installed with a plug-in camera and then print out the tickets immediately. There are four major stages in the E-Bill system: capture the image, extract the license plate in the image, segment the license numbers and recognize the numbers. Extraction and segmentation are elaborately processed by some well-known technologies of digital image processing. Using the back propagation neural network to perform the recognition is the very contribution of the system. Experimental results show that the E-Bill system can effectively recognize most of Taiwan's license plates, which include 10 digits and 26 alphabets. The recognition rate is 90% at a low resolution of 1.3M pixels. The recognition time takes 2 seconds on PDAs and less than 10^{-3} second on personal computers. After empirically evaluation, we can conclude that the E-Bill system is quite impressive with its excellent portability and still has a competitive performance even under its inherent limitations. From a practical point of view, we also believe that the E-Bill system is worth introducing to many parking areas or other application domains.*

Keywords: License plate recognition, back-propagation neural network, digital image processing.

1. Introduction

License plate recognition is an area of long-standing and ongoing research and still receives urgent attention on usable and practical solutions. Limited by their portability, however, the previous systems were usually installed in places of interest to identify vehicles that violate traffic laws or to find stolen vehicles. With the advent of personal digital assistants (PDAs), license plate recognition system finds wider varieties of places to fit itself beyond just controlling access to a toll collection point or parking lot. For example, parallel parking fee collection involves some issues to be overcome. First, the cars can be parked, without a single point of entrance, anywhere in the permitted area. No places are adequate to install cameras. Second, the recognition should be fast and accurate enough after taking an image of the car. In such parking area of Taiwan, all parking clerks have to write down the parking bills in handwriting, including license plate numbers and parking time span. This kind of tedious work motivates us to propose a mobile electronic billing (E-bill) system so that the parking bill can be printed out automatically after a parking clerk takes a picture of the car.

As most of the systems, our E-Bill system is divided into four major stages: image acquisition, license plate extraction, license number segmentation, and license number recognition. Taking advantages of the mobility and portability of PDAs, the system is equipped with a plug-in CMOS camera to capture a car image, recognize its license number and then print out the parking bill immediately. Algorithmic improvements to previous versions of similar system should be made to reach this goal. We first utilize the technologies of digital image processing (DIP) to analyze the whole image, locate the license plate and then segment the license numbers into blocks. Each block represents a number (character) to be recognized. Error correction should be made based on our proposed rules in order to prune the suspicious blocks generated by some inevitable side effects during previous steps. Afterwards, we recognize the license plate number by integrating the following four methods frequently used in pattern recognition: feature matching [1], template matching [3], histogram matching [4] and neural network [2,6]. The experimental results will show the promising effectiveness with the integration of these technologies.

This paper is organized as follows. The system architecture is given is Section 2. Algorithms for the proposed system are described in Section 3.

Experimental results are presented in Section 4. Section 5 concludes this paper.

2. System Architecture

The proposed E-bill system is divided into four major stages: image acquisition, license plate extraction, license number segmentation and license number recognition. Figure 1 shows the skeleton of the proposed system. Image acquisition is the first step of the system. Different image acquisition methods will result in different resolution of images. License plate extraction locates the license plate in the captured image. Afterwards, license number segmentation isolates individual characters from the identified plate. Finally, we will get several segmented blocks called patterns. Each pattern represents a single number (character) to be recognized in the next stage. The final stage is responsible for recognizing the patterns. There have been many papers trying to solve this problem by a variety of methods. The back propagation neural network, BPNN, is one of the well-known technologies, which is used in our proposed model. It has several advantages, including quick inference procedure, simple model, higher accuracy, and so on. In the following section, we will briefly introduce the algorithms we use in each stage of the proposed system.

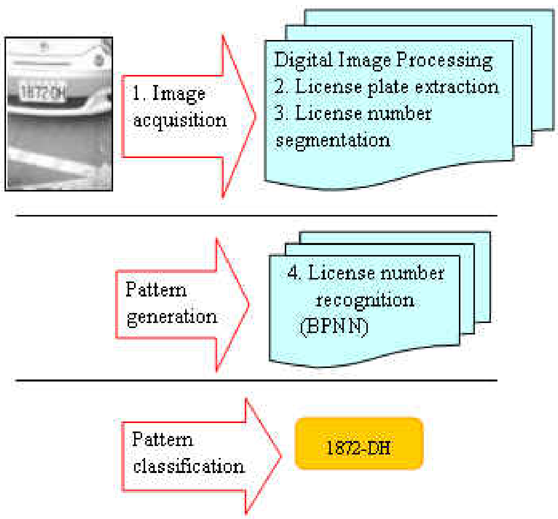

Figure 1. The skeleton of the proposed system.

3. The Proposed Algorithm

Most of well-known algorithms follow similar strategies. The differences between each other are usually at individual detailed processing. There are still many on-going researches in field of image processing [5], pattern extraction [1], and recognition methodologies [7-8]. In our system, we use a PDA installed with FlyCAM CMOS camera to capture images with 1.3 Mega pixels, which is the best quality we can get from the current market. Due to this inherent limitation, we have to put more stress on the image with the technologies of image processing. Figure 2 shows the flowchart of the system. We will get into more details of it in the subsections.

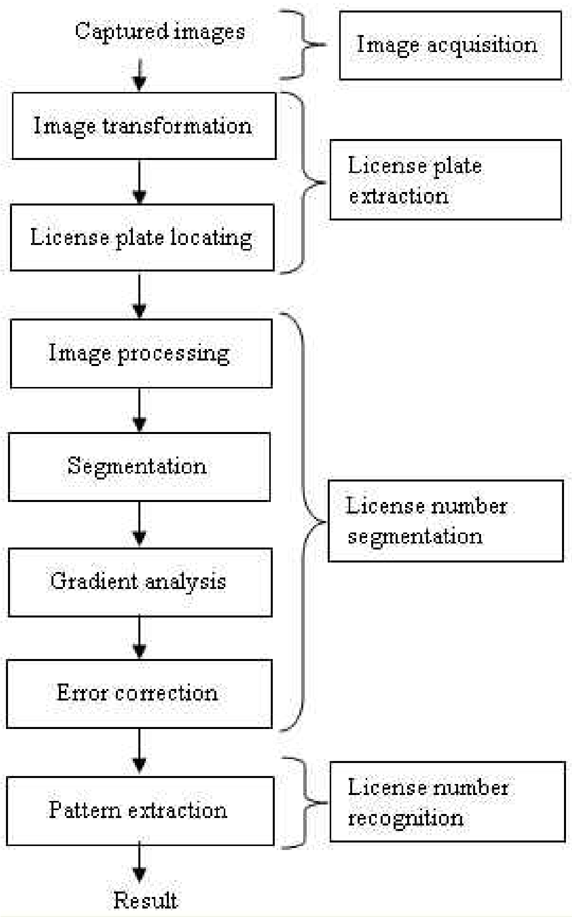

Figure 2. The flowchart of the system.

3.1 Image acquisition

This is the first stage in the system. Almost all the currently running systems use a high resolution digital camera to acquire the image for better recognition accuracy. Instead of using a high-resolution digital camera, the system uses a PDA installed with FlyCAM CMOS camera to capture images for its great portability. The parking clerk can take the system with him/her, patrolling along the road or billing the car. Images are taken in front or rear of the car. The system will store the image files for further processing.

3.2 License plate extraction

License plate extraction is the key step in a license plate recognition system, which influences the accuracy of the system significantly [3]. The system should locate the area of the license plate and then extract it from the image for further processing. There have been many approaches for the extraction of the license plate. For example, the technique in [11] is focused on searching rectangular shape and the background color of car license plate which is brighter than characters. The proposed system calculates the change rate in the predefined rectangle to detect the license plate. The following subsections will illustrate the steps to be taken in this stage.

3.2.1 Image transformation The image obtained from FlyCAM CMOS camera was originally in 24-bit RGB color format. The system will transform it into an 8-bit gray image with size of 120 x 160, which is the standard format used in the system. This image is generated by a popular transformation equation, which preserves the illumination information of the color image while reducing the image complexity.

$$\text{Gray} = R*0.3+G*0.59+B*0.1. \qquad (1)$$

3.2.2 License plate locating First of all, we define a rectangle with size of 30x60, which has the same ratio as the actual license plate. Then, we scan the whole image by using this rectangle and calculate the change rate within this rectangle. The change rate is defined as follows.

$$changed\_rate = \frac{change\_count}{area}, \qquad (2)$$

where change_count is increased by one when the gray values of two adjacent pixels change from 255 to 0, or from 0 to 255. The rectangle with highest value is assumed to be the license plate. Notice that the captured image might have different size. A little processing, either stretching or shrinking, has to be automatically done here. Figure 3 shows some examples of extracted license plates.

Figure 3. Some examples of extracted license plates.

3.3 License number segmentation

Individual numbers (characters) should be isolated from the extracted rectangle before they can be recognized. It is preferable to divide the rectangle into six numbers because Taiwan's license plate is in form of DDDD-XX or XX-DDDD, where X stands for numeral or letter and D for numeral. For example, 9275-CD, 4262-D3, XD-4305 and V2-7841 are all legal format of license plates. Without the prior knowledge of character, our system faces more challenge than the system in [4].

3.3.1 Image processing Through enhancing the image, we can increase its resolution, which does not add any additional information, but is helpful for the subsequent steps. The system will first strengthen the slight differences between two close pixels in the rectangle. The effects of such enhancement are shown in Figure 4. We can see that Figure 4(b) is clearer than Figure 4(a). Then, we magnify the rectangle twice to increase the gap between two consecutive numbers. In each time, we get an image with 1.5 times larger in size. A linear interpolation method can be used as a low-pass filter here. Finally, Laplacian operator is applied to sharpening the edge of object in an image. The following is used as mask in the system.

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{pmatrix}.$$

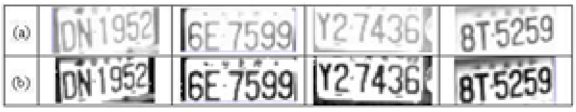

Figure 4. (a) Original plate images. (b) Results after enhancement.

3.3.2 Segmentation Before segmentation, the enhanced images need to be binarized. Several binarization algorithms have been developed by some computer vision and image processing communities. There were some discussions in [9] on the formula used in calculating the threshold. Our system, however, simply binarizes images with threshold t = 128. Consequently, image pixels have only two gray levels, i.e., 0 and 255. The adjacent pixels of the same gray level were connected and assigned a unified number. Afterwards, we will get several blocks, each of which corresponds to a number of the plate. We are expected to get six blocks. But sometimes we will probably have more. Some of them may just represent noise segments and need to be verified and disposed.

3.3.3 Gradient analysis Among the segmented blocks, we choose the middle one as the standard pattern and record its height-width ratio. The blocks with height-width ratio far different from the standard one are discarded. Then, gradient analysis is utilized to find three lines crossing through the remaining blocks. Figure 5(a) and Figure 5(b) show four images and the results after gradient analysis, respectively. We can see that the straight lines reveal the alignment trend of the blocks. It is important for the system to detect noise blocks while trying to conduct error correction.

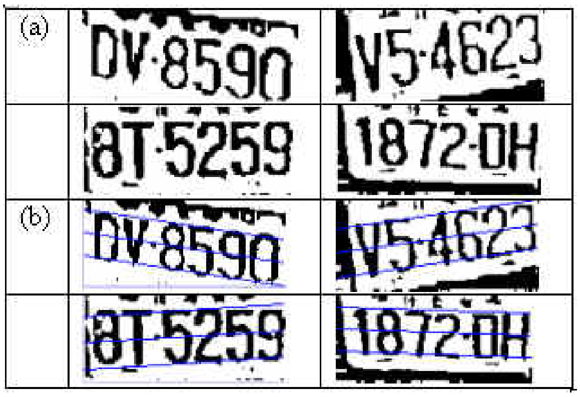

Figure 5. The results after gradient analysis.

3.3.4 Error correction Due to the image size, brightness, or some side effects after image processing, there might be some unexpected branches connected to a license plate number. We develop the following algorithm to conduct error correction.

```
// Steps used to conduct error correction
Determine the standard pattern size
Compare each pattern with standard pattern and
store their condition, where Condition =
{small_height, large_height, small_width,
large_width }
For each pattern
   if (small_height & small_width)   // rule 1
      {discard this block;}
   else if (large_height & large_width)  // rule 2
      {discard this block;}
   else if (large_height & small_width) // rule 3
      {discard this block;}
   else if (large_width)   // rule 4.
      {split the block; }
   else if (large_height)   // rule 5
      {adjust the height and add to the list;}
   else {correct block, add to the list directly; }
End for
```

Rule one prunes small blocks, such as dash or minor noise. The second rule avoids larger area, such as the noise generated from license plate's borders. The third rule discards the border block which might be regarded as number "1" or character "I". The fourth rule is responsible for separating two connected blocks. The fifth rule is used to prune the vertical branch of a block. Figure 6(a) gives original images, Figure 6(b) shows the segmented images without error correction, and Figure 6(c) presents the final segmented images after error correction. From the observation, compared with Figure 6(b), Figure 6(c) has successfully removed some noise as expected.

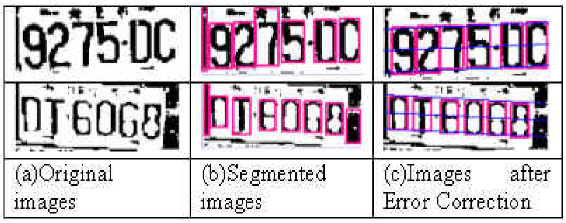

Figure 6. (a) The original license plate images, (b) the segmented images and (c) the images after error correction.

4. Experimental Results

The E-Bill system is implemented on a PDA with Intel PXA255 300MHz CPU and 64M built-in memory, using MS Evc 3.0. It is equipped with FlyCAM CMOS camera to capture images and a portable printer for printing the bills (See Figure 7 and Figure 8). Note that the resolution of the image is only 1.3M pixels, which is much less than those in many currently running systems.

The final system would use a CMOS camera to take a snapshot of the car, process the image file, segment and recognize individual characters, and then print out the parking bill after a user's confirmation.

Figure 7. A glance of the E-bill system.

Figure 8. The portable printer used in the system.

4.1 Training the system

We choose the artificial neural network (ANN) technology as the core of pattern recognition for two reasons. First, it is inherently data-driven – it learns directly from examples of the kind of data it must ultimately classify. Second, ANNs can carve up the sample space effectively, with nonlinear decision boundaries that yield excellent generalization, given sufficient training data [12]. Among the well-known methods, the back-propagation (BP) model is the most common learning algorithm. The ANN that utilizes the BP model is called a BPNN, which performs a pattern recognition task in the proposed system.

There are 36 patterns to be recognized in the BPNN, including 10 numerals and 26 letters. The recognition performance is found to be greatly influenced by the training patterns. For each pattern, therefore, we use 9 kinds of test font to train the network such that a pattern can still be recognized even it is somewhat slant or deformed. Figure 9 shows 9 test instances of the numeral “0”. This is necessary to make the recognition results less sensitive to the variation of images.

There are 150 input nodes, 700 hidden nodes and 36 output nodes used in the BPNN. Training is completed when the error is below 0.001. Then, all the weights and parameters are stored in the system.

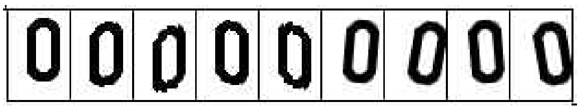

Figure 9. Nine test fonts for number “0”.

4.2 The experimental results

Input of the system is a sequence of color images acquired by camera and its output is a candidate image waiting for users’ further confirmation. In order to evaluate the proposed methods, a set of tests has been performed. Three hundreds of images are taken in an open-space parking area under various illumination conditions, such as at noon, in the evening, in rainy and cloudy days, etc. Moreover, a shape of a license plate observed in an image depends on the relative placement between the camera and cars [10]. Therefore, some of them are even intentionally taken from different angles, a little slant from left, right, up and down sides. The objective is to test the limitation of the slant angle that the license plate could be. The system also provides user some optional candidates for further verification, avoiding possible misjudgments because of their similar appearances such as 8 and B, zero and O, 1 and I, etc. Figure 10 is a snapshot of the system, simulated in a personal computer equipped with PIII 700 CPU and 128M DRAM.

The recognition rate is 90% under the most rugged conditions, like 20 degree of slant angle, low resolution of images, and cloudy weather. Actually, the bad quality of input image is the main reason that causes the system fail to recognize it. The error correction gives good results, ruling out unclear detection or extraction of the edges.

Figure 10. The recognition of an image.

5. Conclusion and Future Work

To our best knowledge, the previous license plate recognition systems took pictures at fixed places with high-resolution digital camera. However, our system is built on a PDA and uses a plug-in FlyCAM CMOS camera to capture images. That is, the system takes advantages of its portability at the cost of low resolution.

Generally, the whole system is working fine achieving most users’ attention for its great portability. The recognition rate is 90% for this version and seems lower than other systems. This is because the images are only 1.3M pixels, and some of them are taken from different angles. Moreover, the format of Taiwan’s license plate is another issue. A letter or numeral could appear at any digit of the license plate, which makes the system easy to make misjudgments for 8 and B, zero and O, 1 and I. Instead of giving one affirmative answer, the system outputs some candidate numbers with the most similar one on top so that the system does not have to spend too much time in distinguishing them.

The experimental E-Bill system is divided into four major stages: image acquisition, license plate extraction, license number segmentation, and license number recognition. Some well-known digital image processing technologies are well utilized to process the first three stages. Our proposed error correction algorithm is proved to be useful in eliminating ambiguity while conducting the segmentation. It is observed that the well-trained BPNN also provides a high level of accuracy for recognizing the patterns in the final stage even though the input patterns are somewhat distorted.

In summary, experimental results show that our system has great portability under an acceptable accurate rate. This could encourage a lot of thoughts in practical domains, such as billing parked cars, finding stolen cars in an open space, enforcing security control, etc. Based on this promising result, we will move forward to the next phase of the research. First, more intensive works have to be done in cleaning up the image, such as background removal, noise filtering, etc. Second, more diverse experiments should be made to improve five rules of error corrections, which are of great help in handling lower resolution images.

Acknowledgment

This work is supported by National Science Council, Taiwan, R.O.C. under Grants NSC92-2213-E-036-017, NSC92-2516-S-036-001, and by Tatung University under Grant B9208-I02-025.

REFERENCES

[1] J. Barroso, E.L. Dagless, A. Rafael and J. Bulas-Cruz, “Number plate reading using computer vision,” *IEEE Int. Symp. on Industrial Electronics*, vol. 3, pp.761-766, July 1997.

[2] K.K. Kim, K.I. Kim, J.B. Kim and H.J. Kim, “Learning-based approach for license plate recognition,” *IEEE Signal Processing Society Workshop Neural Networks for Signal Processing X*, vol. 2, no. 11-13, pp.614-623, Dec. 2000.

[3] M. Sarfraz, M.J. Ahmed and S.A.Ghazi, “Saudi Arabian license plate recognition system,” *Proc. of Int. Conf. on Geometric Modeling and Graphics*, pp.36-41, July 2003.

[4] C.A. Rahman, W. Badawy and A. Radmanesh, “A real time vehicle's license plate recognition system,” *Proc. of IEEE Conf. on Advanced Video and Signal Based Surveillance*, pp.163-166, July 2003.

[5] T. Jinshan, E. Peli and S. Acton, “Image enhancement using a contrast measure in the compressed domain,” *IEEE Signal Processing Letters*, vol. 10, no. 10, pp.289-292, Oct. 2003.

[6] H.C. Lui, C.M. Lee and G. Fang, “Neural network application to container number recognition,” *Proc. of Computer Software and Applications Conference*, pp.190-195, Nov. 1990.

[7] J.C.M. Lee, “Automatic character recognition for moving and stationary vehicles and containers in real-life images,” *Int. Joint Conf. on Neural Networks*, vol. 4, pp.2824-2828, July 1999.

[8] J.H. Park and V. Govindaraju, “Active character recognition using “A*-like” algorithm,” *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp.82-87, June 2000.

[9] C. Wolf, J.M. Jolion and F. Chassaing, “Text localization, enhancement and binarization in multimedia documents,” *Proc. of 16th Int. Conf. on Pattern Recognition*, vol. 2, pp.1037-1040, Aug. 2002.

[10] T. Naito, T. Tsukada, K. Yamada, K. Kozuka and S. Yamamoto, “License plate recognition method for inclined plates outdoors,” *Proc. of Int. Conf. on Information Intelligence and Systems*, pp.304-312, Oct. 1999.

[11] T. Sirithinaphong and K. Chamnongthai, “Extraction of car license plate using motor vehicle regulation and character pattern recognition,**”** *The 1998 IEEE Asia-Pacific Conf. on Circuits and Systems*, pp.559-562, Nov. 1998.

[12] L. Yaeger, “Neural networks provide robust character recognition for Newton PDAs,” *IEEE Expert*, pp.10-12, Aug. 1996.