Draft Version
Published in International Symposium on Signal Processing and its Applications (ISSPA'07), Sharja (U.A.E.), 2007.

A VERSATILE LOW-COST CAR PLATE RECOGNITION SYSTEM

ABSTRACT

The utility of automatic systems to visually identify vehicles based on the recognition of their license plates is nowadays unquestionable. They can be applied in very different scenarios, like access control, calculation of parking fares, automatic payment of tolls or parking fines, traffic control, etc. In the literature numerous works can be found proposing solutions to the automatic license plate recognition (LPR). Nevertheless most works propose specific LPR systems for particular applications, imposing applicability restrictions that limit their use. Restrictions range from the assumption of stationary backgrounds and fixed lighting conditions to a limited (or null) speed for the vehicles. These restrictions are sometimes overcome through expensive and complex systems. This paper proposes a versatile LPR system that yields good results running on a low-cost platform in non-controlled environments. Our method, based on multiple neural networks, is fast enough to be applicable in camera-in-motion applications and exhibits a high rate of success (up to 95%) in scenarios with non-stationary backgrounds, and varying lighting conditions.

1. INTRODUCTION

The automatic license plate recognition (LPR) is a field of high interest and deeply studied. There is a vast literature addressing this problem given the practical and economical benefits yielded by LPR systems and their utility in a wide range of applications, including automatic toll payment [2],[5], access control to buildings and facilities [4], for imposing parking fines or congestion taxes [3], etc.
Most systems exploit the particular conditions of the application to restrict their works to static backgrounds [8] or to controlled light scenes [5]. In some cases, additional (and usually expensive) components like infrared cameras are considered to improve the recognition performance and/or to fix the distance between the camera and the vehicle [6],[7].
In this work we present a versatile automatic LPR that can perform well on a low-cost platform, i.e. a laptop, using images acquired from a conventional webcam. Our system is capable of working in a variety of scenarios and conditions (variable light conditions, background and position of the vehicle) without requiring the adjustment of any critical parameter. We have devised an algorithm that robustly localizes in the image plates of different sizes (i.e., distance to the vehicle) and orientations. The posterior character recognition over the segmented region is carried out through the combination of multiple neural networks that provides excellent results (around 95% of success). The system is being working from the last year on a control access application in our University at the entrance of the Computer Science building.
The rest of the paper is organized as follows. Section 2 describes, in general lines, the proposed LPR method, which is detailed in sections 3 and 4. Section 5 presents results of our approach in different situations. Finally, some conclusions and future work are outlined.

2. THE PROPOSED LPR APPROACH

In any automatic plate recognition system, two different phases can be distinguished. First, a particular region within the input image has to be identified as a car plate (*localization*), and then a character sequence inside the region has to be validated as a correct plate string following some grammatical rules (*recognition*).
Obviously, an accurate and reliable localization is crucial for the success of the plate recognition. Assuming non-controlled conditions, the localization process should face a variety of situations like locating plates at different distances, with different orientations, light conditions, etc. On the other hand, character recognition is neither an easy problem: depending on the scene conditions, some characters are hardly recognizable or distinguishable from others.
Our overall approach, depicted in figure 1, considers a localization method (explained in section 3) that locates characters within greyscale images, eliminates the scene background, and corrects the image to obtain a rectified picture of the detected car plate. After that, we rely on a set of redundant neural networks for the recognition phase (analysed in section 4), trained with different pattern sets to identify alphanumeric characters. Additionally, some networks have been specifically designed to disambiguate between pairs of problematic characters, such as ‘8’ and ‘B’. This disambiguation stage is also considered in other works as in [9], in which a minor comparison between them is performed focusing only on the non-ambiguous parts of the characters.
A distinctive characteristic of our approach is that the entire process is repeated for different scales of the input image (which is resampled) in order to make the system more robust and versatile, working with a wide range of distances between the vehicle and the camera. The range

of distances for which our current setting (algorithm and camera) performs well is between 50 cm. to 5 m., though this may be reconfigurable if a camera with more resolution is used.

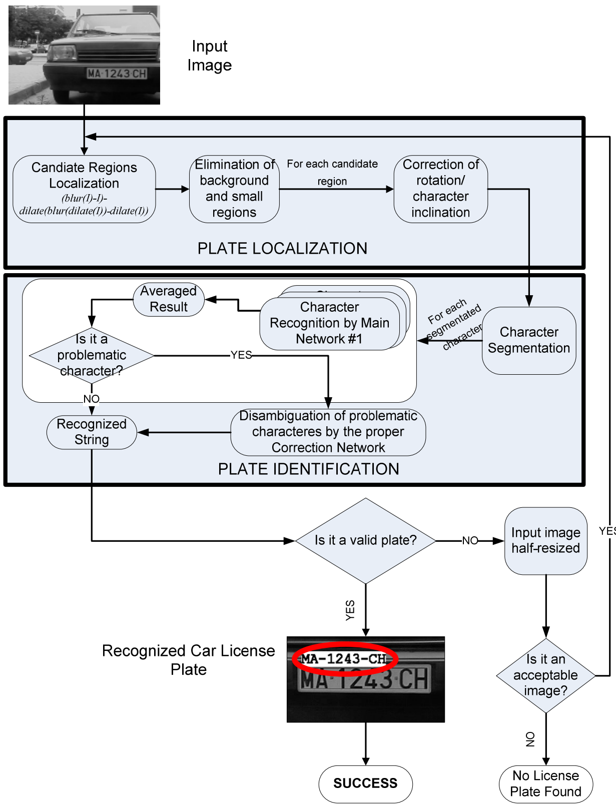

Figure 1. Scheme of our LPR system. The localization and recognition stages are successively repeated for different sizes of the input image if needed. This increases the system robustness with respect to the distance between the vehicle and the camera.

3. THE PLATE LOCALIZATION STAGE

The proposed method for plate localization is based on the identification of groups of dark strokes on lighter regions. This approach, applied at different scales of the input image, has demonstrated to be barely affected by the plate distance and works well with different orientations. More specifically, our method finds dark strokes on the image by applying the following operator:

$$\left[blur_{7x7}(I)-I\right]-dilate\left\{blur_{7x7}(dilate(I))-dilate(I)\right\} \tag{1}$$

Since the thickness of the plate's characters is almost constant, operation (1) is able to keep the plate's strokes (for a particular scale of the input image) while getting rid of those of a very different thickness, as shown in figure 2b. Next, after binarizing the image, isolated and sparse groups of pixels are eliminated based on their local density (figure 2d and 2e). Though some characters can be partially removed or attenuated during these operations, it does not affect in a great extent the overall localization stage since, at this point, our purpose is not to segment the exact plate, but to identify the positions of candidate plates in the image (its precise segmentation will be commented further on). In the event that several regions remain after this stage, small ones are discarded and the rest (typically no more than two) are all processed through the following stages, selecting at the end the one with the highest recognition confidence.

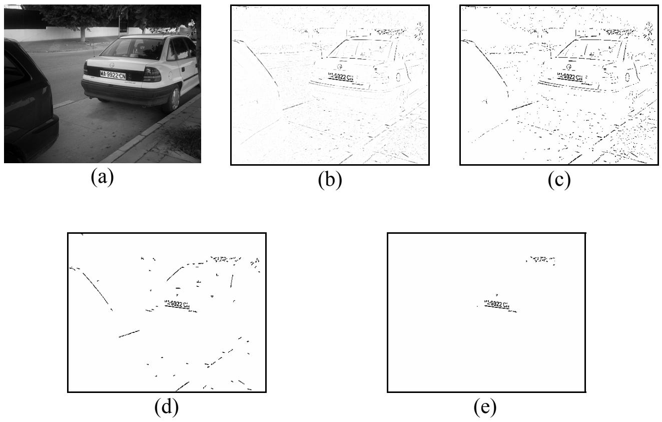

Figure 2. An example of the localization stage. a) Input image. b) Results of the application of (1). c) Binarization of image b. d), e) Elimination of isolated and sparse pixels.

In general, plates may appear distorted by a projective or affine transformation which would require the identification in the image of 4 or 3 points, respectively, of known 3D relative geometry (e.g., the corners of the plate). Although in theory this correction is quite simple, in practice, searching for those points may become inefficient and prone to errors in their localization. An alternative way of correcting the affine distortion is proposed in [1], where the Radon Transform is utilized, though its computational burden is still a problem that limits the possible plate orientation in the input image to the range of $\pm 8^{o}$. To overcome this, we perform a two-step orientation correction, first for the possible rotations of the plate, and then for the characters inclination. For finding out these two correction angles, we make use of lateral histograms, both horizontal and vertical. A lateral histogram represents the number of black (or white) pixels along a given image line (typically, rows and columns for horizontal and vertical histograms, respectively). Thus, to correct the plate orientation, the horizontal histogram of the candidate region is computed for different rotation angles ($\pm 30^{o}$) (see figure 3-left). The angle θ for which the histogram yields the highest variance σ indicates the horizontal orientation of the plate in the image:

$$\sigma_{\theta}=\frac{1}{N}\sum_{i=1}^{N}\left(h_{\theta i}-\overline{h_{\theta}}\right)^{2} \tag{2}$$

where $h_{\theta i}$ is the value of the histogram for the image row i, $\overline{h_\theta}$ the mean value of $h_{\theta i}$, and N is the number of rows in the rotated plate image.
The width of the central subhistogram indicates the height of the string characters in the plate. The candidate region is now expanded proportionally to that height along the computed orientation (for Spanish license plates the height-length ratio is 1:5).

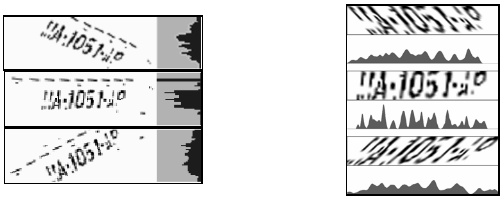

Figure 3. Preprocessing of a plate candidate. Left) Correction for the plate rotation. The lateral histogram provides information about the rotation correction to be applied. Right) Correction of character inclination. For different inclinations, the lateral histogram with the highest number of gaps yields the proper correction of characters.

Once the candidate region is expanded and rotated, the vertical histogram is calculated for different angles to correct the possible character inclination (see figure 3-right). The proper correction is the one that produces the highest variance σ in the histogram. Gaps in this histogram represent blanks in the alphanumeric sequence, are also employed to segment the plate characters.

4. THE PLATE RECOGNITION STAGE

Once a candidate plate region has been segmented within the input image, the next step is the segmentation of the characters and its recognition and validation of its characters. In the literature, different techniques have been proposed for character recognition within LPR systems (see [9] for some related works). Our approach relies on the use of neural networks (one of the most commonly adopted solutions in this field) given their low computational burden. To improve recognition robustness, we implement a set of multilayer feedforward neural networks grouped in two classes: *main nets* in charge of recognizing the whole set of alphanumeric characters and *correction nets* devoted to disambiguate some problematic characters like '8'/'B', 'O'/'D', and 'I'/'1'. The input for both types of networks is a sequence of character images, segmented from the information provided by the gaps detected in the vertical histogram (explained above) and resized to a fixed scale (not explained here for lack of space).
In our particular implementation we use 6 main networks that contain a number of neurons at the hidden layer that varies from 16 to 64 (see figure 4).

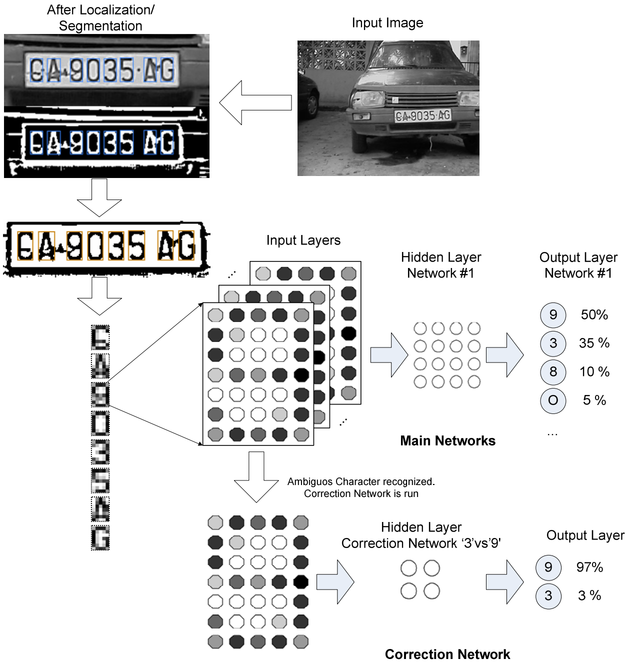

Figure 4. Representation of main networks (top) and a correction one (bottom). The number of input neurons is given by the size of the characters (5x7). The number of output neurons is 35 for main nets and two for correction nets.

Each neural network has been trained with a different set of character patterns which were generated by adding random deformations and noise to a synthetic ideal characters set. Results from main networks are averaged to yield the character identified with the highest confidence. This redundant scheme has experimentally demonstrated to provide more robust and reliable results than using only one network for character recognition.
When the character proposed by the main networks belongs to the predefined set of problematic characters ('3'/'9', '8'/'B', 'O'/'D', or 'I'/'1') a specific correction net, specially trained to disambiguate between them, is executed.
Finally, a syntax verification is carried out considering the two plate formats that coexist in Spain: '1234-AAA' and 'A[A]-1234-A[A]'.

5. EXPERIENCES AND RESULTS

The developed LPR system has been tested in two different configurations: in a car control access application, and for an in-movement recognition application.
In the control access application, (see figure 5), a conventional IP camera is used to transmit image sequences to a host computer which performs the license recognition. In this application, our system yields a 95% of success (calculated with an average of 50 vehicles per day), being the errors mainly due to failures when processing deteriorated plates. The system works in a dynamic scenario (in presence of people and moving cars

in the surroundings) during the whole working day (8am-8pm) with natural and artificial light (a near streetlamp is on at night).

Figure 5. Left) A real implementation of our system for controlling the access to our building. Right) A snapshot of the software application.

To test the suitability of the proposed system in non-stationary applications and more variable conditions, we have conducted several experiments consisting of identifying plates in an outdoor parking, setting up our system with a laptop and a webcam in a vehicle in motion (see figure 6). In these tests, the front passenger aims the webcam to the parked cars while our vehicle is moving through the parking area. The success rate in this case decreases to 80%, mainly due to the low performance of the webcam for capturing images in motion and adapting to variable lighting conditions.

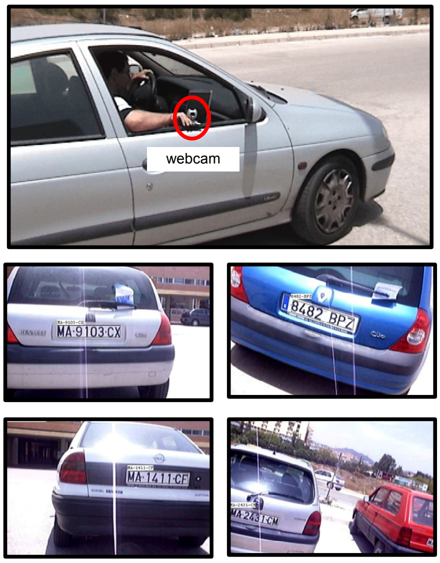

Figure 6. In-motion application of the proposed LPR system. An operator aims a webcam toward the parked vehicles.

Regarding the efficiency of our system, which has been optimized through the Intel® Image Processing Library (IPL), the overall process takes around 90 ms. (aprox. 80% in localization and 20% for recognition) for 640x480 b&w image in a Pentium IV Prescott HT at 2.8 Ghz. These figures permit us to process up to 10 images per second, enabling the system to repeat the entire plate identification process a number of times before ending up with a result, improving, thus, the system robustness (in our implementation a valid output is produced after three consecutive executions with identical result).

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a LPR system aimed to face a variety of situations that commonly restrict other works, i.e. variable background, light conditions, plate orientations, etc.

Our experience with the proposed system has demonstrated its versatility and robustness, even in very demanding applications where the plate may be captured at different distances and orientation, and also with arbitrary backgrounds and different illuminations of the scene.

Future works considers the recognition of foreign license plates and the implementation of our approach in a PDA for urban traffic fining.

REFERENCES

[1] V. Shapiro, G. Gluhchev and D. Dimov. "Towards a multinational car license plate recognition system", *Machine Vision and Applications*, vol. 17, pp. 173-183, May 2006.

[2] R.A. Lotufo, A.D. Morgan and A.S. Johnson, "Automatic number plate recognition" in *Proc. Int. Elect. Eng. Colloquium on Image Analysis for Transp. Applications,* pp. 6/1-6/6, 1990.

[3] J.R. Cowell, "Syntactic pattern recognizer for vehicle identification numbrs", *Image and Vision Comput.*, vol. 13, no. 1, pp. 13-19, 1995.

[4] S. Draghici, "A neural network based artificial vision system for license plate recognition", *Int. J. Neural Systems*, vol. 8, pp. 113-126, 1997.

[5] P. Davies, N. Emmott, and N. Ayland, "License plate recognition technology for toll violation enforcement" in *Proc. Int. Elect. Eng. Colloquium on Image Analysis for Transport Applications,* pp. 7/1-7/5, 1990.

[6] S.H. Lee, Y.S. Seok, and E.J. Lee. "Multi-national integrated car license plate recognition system using geometrical feature and hybrid pattern vector", in *Proc. ITC Conf. Circuits/Systems Computers and Comm. Phuket, Thailand.*, pp. 1256-1259, 2002.

[7] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust license-plate recognition method for passing vehicles Ander outside environment" *IEEE Trans. Veh. Technol.,* vol. 49, pp. 2309-2319, 2000.

[8] L. Salgado, J.M. Menendez, E. Rendon, and N. Garcia, "Automatic car plate detection and recognition through intelligent vision engineering", in *Proc. IEEE Int. Carnahan Conf. Security Technology,* pp. 71-76, 1999.

[9] S-L. Chang, L-S. Chen, Y-C. Chung and S-W. Chen. "Automatic License Plate Recognition" *IEEE Trans. on Intell. Transportation Systems*, vol. 5, no. 1, pp. 42-53, 2004.