Multiple clues for license plate detection and recognition

Pablo Negri[1], Mariano Tepper[2], Daniel Acevedo[2], Julio Jacobo[2], and Marta Mejail[2]

[1] PLADEMA-UNICEN, Tandil, Argentina
http://www.pladema.net
[2] Departamento de Computación-UBA, Buenos Aires, Argentina
http://www-2.dc.uba.ar/grupinv/imagenes/

Abstract. This paper addresses a license plate detection and recognition (LPR) task on still images of trucks. The main contribution of our LPR system is the fusion of different segmentation algorithms used to improve the license plate detection. We also compare the performance of two kinds of classifiers for optical character recognition (OCR): one based on the *a contrario* framework using the shape contexts as features and the other based on a SVM classifier using the intensity pixel values as features.

1 Introduction

License plate recognition (LPR) currently finds other applications than the electronic payment systems (toll payment and parking fee payment) or traffic surveillance. Entrepreneurs discover the usefulness of identifying their clients, for example, by using this technology to study clients' shopping habits in a fast food drive-thru. The present work involves an application where the LPR solution is installed on a truck balance to identify incoming vehicles making it possible to record the truck's weight automatically.

Figure 1 shows some samples of the images captured by the camera system. The system has been installed outdoors and works during day and night. It can be seen from the samples that the distance between the camera and the vehicle is variable, and the license plate can be anywhere in the image. Finally, characters in the license plate can be distorted, noisy, broken or incomplete, challenging the simple methods used in the commercial systems.

The state of art in LPR systems is well summerized in the work of Anagnostopoulos [1]. They present the LPR algorithm as a three-step framework: 1) LP location; 2) characters segmentation; and 3) character recognition. In general, the first step should operate fast enough to fulfill the need of real time operations. For still images, which are the scope of our work, methods in the literature include techniques that take advantage of the high contrast in the license plate: morphological operations, edge detection [10], hierarchical representations [4], image transformations [7], etc. There are also detection algorithms based on

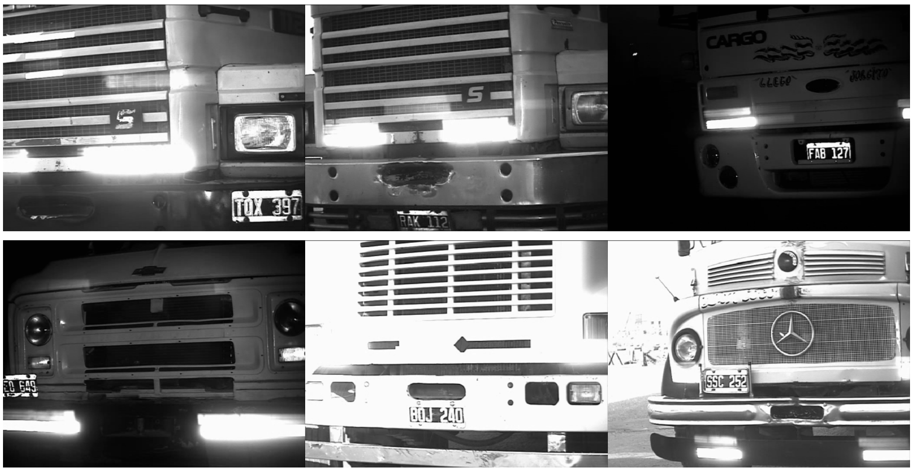


Fig. 1: Truck image samples with different license plate positions.

AdaBoost [8] and support vector machine (SVM) [9] classifiers, using Haar-like features or the color and texture information.

The character segmentation step examines the potential LP locations to determine the character bounding boxes. The final step matches the extracted characters to a number or a letter. In this stage, different types of classifiers have been applied such as SVM [4], artificial neural networks (ANN) [12], etc.

This paper describes a three steps framework for robust license plate detection and recognition. The main contribution of our LPR system is the fusion of different kinds of segmentation algorithms to obtain a strong license plate detector.

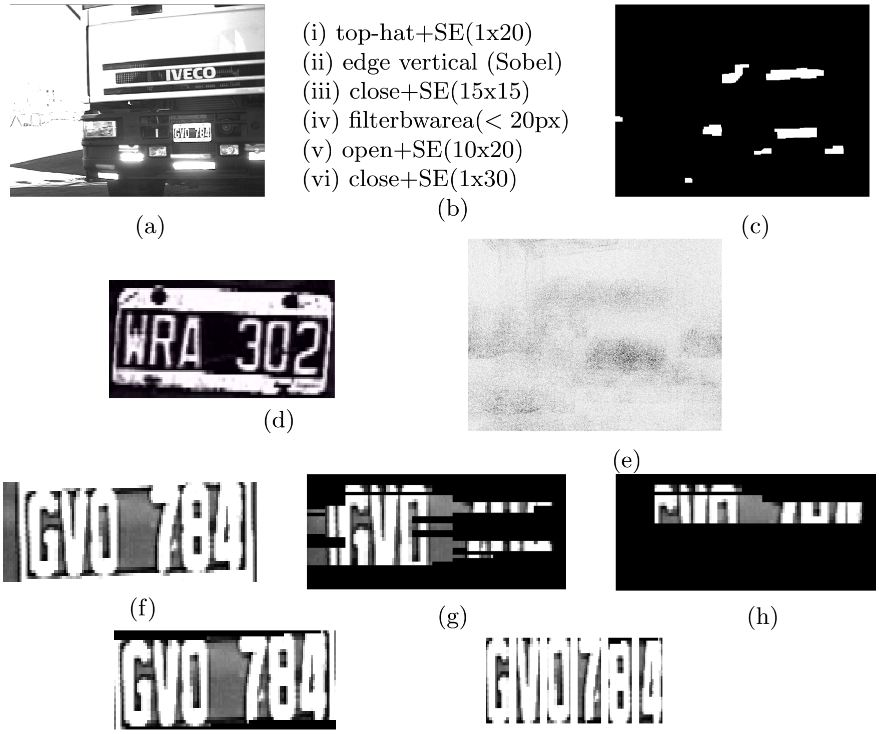
The article is organized as follows: section 2 presents the detection and recognition framework, section 3 shows the results of our system, finalizing with conclusions and perspectives in section 4.

2 LPR Framework

In this section we introduce the three steps of the LPR framework: license plate detection, character segmentation and character recognition.

2.1 License Plate Detection

The first task of a LPR system is the detection of the license plate inside the image. The LP detection process starts generating several Regions of Interest (RoIs) using morphological filters. To validate the RoIs and choose the most probable license plate region, we perform a more exhaustive evaluation: analysis of the presence of text and obtainment of a correlation map using the Fourier



```
(i) top-hat+SE(1x20)
(ii) edge vertical (Sobel)
(iii) close+SE(15x15)
(iv) filterbwarea(< 20px)
(v) open+SE(10x20)
(vi) close+SE(1x30)
```

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 2: License plate detection. (a) Original image; (b) Pseudo code of the morphological operations; (c) resulting RoIs; (d) correlation pattern; (e) correlation map; (f) extracted RoI; (g) text segments; (h) text blocks; (i) choosen RoI and (j) characters extraction.

transform. These clues, being of different nature, help the system to obtain a strong and robust detection.

Morphology. A morphological top-hat filtering is applied to the input image. This step enhances the image regions with great difference in intensity values. Then, a vertical edge map is calculated using a Sobel operator. Successive morphological operations are then applied in order to obtain potential license plate regions. Fig. 2 (b) shows the pseudo-code of the morphological operations. At each step it also shows the structural element employed and its size. Step (iv) means that all the regions with less than 20 pixels are discarded. We obtain N resulting RoIs, Fig. 2 (c), R_i, $i = 1, ..., N$.

Template matching. We calculate a correlation map of the occurrences of a license plate in the image using a FFT. Let be $\mathbf{I}$ the input image and $F(\mathbf{I})$ its FFT, $\mathbf{P}$ the correlation pattern, Fig. 2 (d), and $F(\mathbf{P})$ its FFT. Then, the correlation map M (see Fig. 2 (e)) is given by $M = F^{-1}\left(\frac{F(\mathbf{I})F(\mathbf{P})^*}{\|F(\mathbf{I})\|\|F(\mathbf{P})\|}\right)$ where

F^{-1} is the inverse Fourier transform and F^* the complex conjugate of F. We get then a feature vector $\mathbf{scv} = (scv_1, \ldots, scv_N)$ of confidence values, where $scv_i = \frac{1}{|R_i|} \sum_{(x,y) \in R_i} M(x,y)$ and $|\cdot|$ is the area of R_i.

Text segments. From each region R_i, $i = 1, \ldots, N$, we calculate the potential text segments using Wong's method [5]. For that, the horizontal intensity gradient is calculated using a derivative mask [-1,1]. Then, at each $(x, y) \in R_i$, the Maximum Gradient Difference (MGD) is computed. This value is the difference between the maximum and minimum gradient values inside an horizontal segment s of width $n + 1$ centered at (x, y). In our application $n = 40$, which is slightly longer than the average size of two characters. Usually, text segments have large MGD values. The segments are then filtered preserving those with a MGD value larger than a certain threshold. A value of *200* has empirically been found to be the best choice for this parameter.

Next step gets the number of background-to-text $n_{b-t-t}(s)$ and text-to-background $n_{t-t-b}(s)$ transitions for each segment s, which should be close if s contains text. Also, the mean and variance of the horizontal distances between the background-to-text and text-to-background transitions in every segment s is computed. We define the following two conditions: $C^{(1)}(s) = \{n_{b-t-t} + n_{t-t-b} >$ threshold$\}$ and $C^{(2)}(s)$=$\{$mean and variance of horizontal distances in s are inside a certain range$\}$. Then, $S_i = \{s \in R_i \,/\, C^{(1)}(s) \wedge C^{(2)}(s)$ holds$\}$, are the validated segments for every R_i, $i = 1, \ldots, N$. We define two features vectors: $\mathbf{nts} = (nts_1, \ldots, nts_N)$ and $\mathbf{mgd} = (mgd_1, \ldots, mgd_N)$, where $nts_i = \#S_i$ and $mgd_i = \frac{1}{nts_i} \sum_{s \in S_i} MGD(s)$.

Text blocks. In this step, the text segments validated previously are merged to form text blocks. For each text segment, the mean and the variance of the intensity values in the original image are calculated. Two continuous segments are merged if the mean and the variance are close using a two pass strategy, top-down and bottom-up. We define a text block feature $\mathbf{ror} = (ror_1, \ldots, ror_N)$, where ror_i is the confidence value indicating the RoI occupation ratio defined as $ror_i = \text{area(text blocks } \in R_i)/\text{area}(R_i)$, for $i = 1, \ldots, N$. Fig. 2 (h) presents two text blocks covering 23% of the area region.

Clues combination and decision. We get four feature vectors, **nts**, **mgd**, **ror** and **scv** of different nature, each of them having a confidence value for each RoI. We need to merge their information in order to decide which of the N regions have obtained the highest values in the vectors. To do so, we create four sorting index vectors: $\mathbf{nts^{si}}$, $\mathbf{mgd^{si}}$, $\mathbf{ror^{si}}$ and $\mathbf{scv^{si}}$. These vectors give an index to each R_i that depends on an ascending sorting: the R_i with the lowest value in the feature vector gets index 1, and the R_i with the highest value gets index N. Then, we define a vector **votes** with length N:
$\mathbf{votes}(i) = \mathbf{nts^{si}}(i) + \mathbf{mgd^{si}}(i) + \mathbf{ror^{si}}(i) + \mathbf{scv^{si}}(i)$, for $i = 1, \ldots N$.
The region R_m, with $m = \arg\max_{1 \leq i \leq N} \mathbf{votes}(i)$ is retained as the license plate. R_m will always be in the latest positions in the sorted vectors receiving the greatest votes. See Fig. 2 (i) for the chosen region in our example.

2.2 Character Segmentation

The RoI R_m that detected as the license plate is thresholded in order to obtain a binary image, where high intensity values correspond to the foreground color. To identify the characters, the algorithm groups the connected foreground pixels in regions and calculates its bounding boxes. In order to filter the non-character bounding boxes, the algorithm evaluates the width and height ratio and the spatial position, being also capable to split connected characters. The final bounding boxes will establish the validated characters (see Fig. 2 (j)).

2.3 Classification

A LP is composed of two groups of three characters. The first group of characters are letters and the second group are numbers (see Fig. 2). If the number of segmented characters is at least four, it is possible to associate their index position on the license plate. The bounding boxes are sent to a classifier specialized in letters or a classifier specialized in numbers. Two types of classifiers are compared in this work: an edge based method and a template based method.

Edge based method
The first classification method is based on the work of Tepper et al. [13]. They employ the shape context [2] as descriptor and the *a contrario* framework [3] to perform the shape context matching.

Let $\mathcal{T} = \{t_1, \dots, t_n\}$ be the set of points of the contour obtained using the Canny's algorithm. For each $t_i \in \mathcal{T}$, we model the distribution of the positions of $n-1$ remaining points in $\mathcal{T}$ relative to t_i. We call this distribution the Shape Context of t_i (SC_{t_i}). In order to render the SC_{t_i} useful, we discretize the values in a log-polar space obtaining a coarse histogram. Each bin of the histogram corresponds to a cell on the partition and their value is calculated as the number of edge points lying inside the cell.

We use the *a contrario* framework, developed as part of the Computational Gestalt project (see [3] for a complete description).

Let $\{SC_i | 1 \leq i \leq n\}$ and $\{SC'_j | 1 \leq j \leq m\}$ be two sets of shape contexts from two different shapes. We want to see if both shapes look alike. The distances between SC_i and SC'_j can be seen as observations of a random variable D that follows some unknown random process. Formally, let $\mathcal{F} = \{F^k | 1 \leq k \leq K\}$ be a database of K shapes. For each shape $F^k \in \mathcal{F}$ we have a set $\mathcal{T}^k = \{t^k_j | 1 \leq j \leq n_k\}$ where n_k is the number of points in the shape. Let $SC_{t^k_j}$ be the shape context of t^k_j, $1 \leq j \leq n_k$, $1 \leq k \leq M$. We assume that each shape context is split in C independent features that we denote $SC^{(i)}_{t^k_j}$ with $1 \leq i \leq C$. Let Q be a query shape and q a point of Q. We define

$$d^k_j = \max_{1 \leq i \leq C} d^{k(i)}_j, \text{ where } d^{k(i)}_j = d(SC^{(i)}_q, SC^{(i)}_{t^k_j})$$

where $d(\cdot, \cdot)$ is some appropriately chosen distance. We can state the *a contrario* hypothesis: $\mathcal{H}_0$*: the distances* d^k_j *are observations of identically distributed inde-*

pendent random variables D that follows some stochastic process. The number of false alarms of the pair (q, t_j^k) is $\mathrm{NFA}(q, t_j^k) = N \cdot \prod_{i=1}^{C} P(D \leq d_j^k | \mathcal{H}_0)$, where $N = \sum_{k=1}^{K} n_k$, and $P(D \leq d_j^k | \mathcal{H}_0)$ is the probability of false alarms (further details to obtain P can be founded in [13]). If $\mathrm{NFA}(q, t_j^k) \leq \varepsilon$ then the pair (q, t_j^k) is called ε-meaningful match.

The classifier decides which character from the database corresponds to the query counting the number of ε-meaningful matches between the shape query and each shape from the database. The database shape that produces the biggest number of matches is selected. In case that there are no ε-meaningful matches for any database shape, a no-match decision is returned. We have only one shape in the base for each class.

Template based method
In the template based method, the pixel intensity values of the character feed a classifier trained with the Support Vector Machine (SVM) algorithm.

In this work, we train the SVM using the algorithm proposed for Platt: the Sequential Minimal Optimization (SMO) [11]. SMO is a simple algorithm (the pseudocode is available at [11]) that solves the SVM quadratic problem analytically inside an iterative process. Its advantage lies in the fact that solving the dual maximization problem for two Lagrange multipliers can be done analytically and very quickly.

The strategy for the classification is the *One Against All* approach. We construct N binary SVM classifiers, each of which separates one class from the rest. The positive samples for the k-th SVM classifier correspond to those of the k-th class. The negative samples are the samples of the other classes. In the training phase, the training samples are resized to a pattern of 16x10 pixels and their intensity values normalized between -1 and 1. In the testing phase, an input sample, resized and normalized, is the input of the N classifiers. It will be classified as the class whose classifier produces the highest value.

3 Experimental results

Dataset
The dataset is composed of 623 images captured by an infrared camera system placed at a truck entrance gate. Captures are taken at any time of the day (there are day and night captures). The dataset is split into two databases. The first base is composed of 151 images and is used to extract the training samples for the SVM training and the patterns for the shape context matching. The second base (472 images) is our test database. There are two labels for each test sample indicating the license plate numbers and the nature of the LP: non-deteriorated or deteriorated. License plates that are deformed, noisy, broken or incomplete are labeled as deteriorated. There are 356 non deteriorated samples and 116 deteriorated.

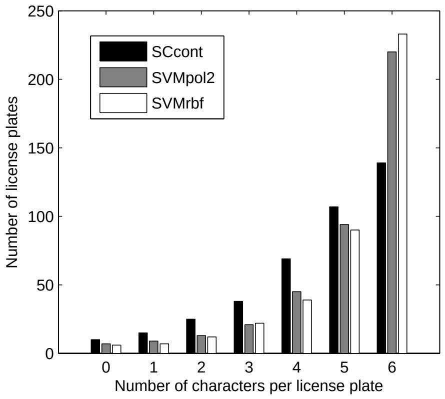
Results
The table in Fig. 3 shows the results of the LPR system. Row **A** presents the

A - Detection		
	Non-det	Det
Detection (%)	94.6	62.0

B - Characters segmentation		
	Non-det	Det
Error (%)	2.2	25.5

C - Classification (%)		
SC*cont*	82.9	68.8
SVMpol2	91.4	85.5
SVMrbf	92.4	90.1

(a)



(b)

Fig. 3: Results of the LPR system. Table (a) shows the detection results in A, error percentage in characters segmentation in B and classification results in C. Figure (b) presents the quantity of characters recognized in the detected LPs.

detection results with a 94.6 % in the non-deteriorated LPs images and 62.0 % for the deteriorated. The average executing time of the detection step using MATLAB running on a PC with 3.16GHz processor is 700 ms. In general, the system misses the samples when there are strong vertical edges in the truck front, which can defeat the LP region in the clues fusion decision. This situation often happens in the case of the deteriorated samples, when the LPs are noisy or deteriorated.

Row **B** exposes the character segmentation errors. We observe the difficult task of this step on the deteriorated images, when there are missing characters or bounding boxes validated out of the boundaries of the LP.

Row **C** exhibits the classification results of the two classifiers on the well extracted characters. For the SVM classifier we use two kind of kernels: SVMpol2, polynomial function (2nd degree) and SVMrbf, radial basis function. The latter obtains the best score. The edge based classifier SC*cont* obtains the worst performance. This is a natural result if we consider that there is only one shape for each class in the database.

Finally, the last bars in Fig. 3 (b) present the number of LPs in which the system recognizes the six characters well. LPs giving five characters means that one character was missed or erroneously classified. Certainly we can expect the LPR system to make some mistakes. In order to improve the system performance, a list of LPs can be used to verify the LPR response and to validate it or change it for the most probable LP in the base.

4 Conclusions and perspectives

This paper presented a LPR system for still outdoor truck images. We obtained good results in the detection phase by using a fusion of different segmentation algorithms. For the recognition phase, we compared an edge based method and a template based method. The former, using the *a contrario* framework and the shape context features, has the advantage that uses only one shape per class. The latter, based on a SVM classifier and pixel values as features, obtained the best performance.

We consider, however, that further research is necessary to tackle deteriorated LP images on the detection and character segmentation phases. Consequently, we plan on applying new clues to minimize erroneous detection results, especially for deteriorated images. As far as it is possible, we expect to make use of the number of segments inside each region extracted using the Hough transform. More detailed research can also be done in the characters segmentation phase in order to improve the performance of the system. Finally, classification results using the edge based method could be improved generating shapes in a different way that better generalize each class.

References

1. Anagnostopoulos, C. et al.: License plate recognition from still images and video sequences: A survey. Inteligent Transportation Systems 9(3), 377–391 (2008)
2. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. Pattern Analysis and Machine Intelligence 24(4), 509–522 (2002)
3. Desolneux, A., Moisan, L., Morel, J.: From Gestalt Theory to Image Analysis: A Probabilistic Approach. Springer (2008)
4. Donoser, M., Arth, C., Bischof, H.: Detecting, tracking and recognizing license plates. In: ACCV. pp. 447–456 (2007)
5. E. K. Wong, M.C.: A new robust algorithm for video text extraction. Pattern Recognition 36, 1397–1406 (2003)
6. Hongliang, B., Changping, L.: A hybrid license plate extraction method based on edge statistics and morphology. In: ICPR. pp. 831–834 (2004)
7. Hsieh, C.T., Juan, Y.S., Hung, K.M.: Multiple license plate detection for complex background. In: AINA. pp. 389–392 (2005)
8. Huaifeng, Z. et al.: Learning-based license plate detection using global and local features. In: ICPR. pp. 1102–1105 (2006)
9. K.I.Kim, Jung, K., Kim, J.: Color Texture-Based Object Detection: An Application to License Plate Localization, vol. 2388, pp. 321–335. Springer (2002)
10. Mello, C.A.B. and Costa, D.C.: A Complete System for Vehicle License Plate Recognition, In: WSSIP, pp. 1–4 (2009).
11. Platt, J.C.: Sequential minimal optimization: A fast algorithm for training support vecthor machines. Tech. Rep. MSR-TR-98-14, Microsoft Research (1998)
12. Rahman, C.A., Badawy, W., Radmanesh, A.: A real time vehicle's license plate recognition system. In: AVSS. pp. 163–166 (2003)
13. Tepper, M., Acevedo, D., Goussies, N., Jacobo, J., Mejail, M.: A decision step for shape context matching. In: ICIP. pp. 409–412 (2009)